Walter Franz
Hannes Hartenstein
Martin Mauve (Eds.)

Inter-Vehicle-Communications Based on Ad Hoc Networking Principles

The FleetNet Project

universitätsverlag karlsruhe

Walter Franz, Hannes Hartenstein, Martin Mauve (Eds.)

Inter-Vehicle-Communications Based on Ad Hoc Networking Principles

The FleetNet Project

Cover photo by DaimlerChrysler Media Services used with permission.

Inter-Vehicle-Communications Based on Ad Hoc Networking Principles

The FleetNet Project

Walter Franz
Hannes Hartenstein
Martin Mauve
(Eds.)

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

universitätsverlag karlsruhe

Impressum

Universitätsverlag Karlsruhe
c/o Universitätsbibliothek
Straße am Forum 2
D-76131 Karlsruhe

www.uvka.de

Print on Demand

ISBN 3-937300-88-0

Preface

Vehicular ad hoc networks (VANETs) can help to increase safety and comfort ‘on the road’. As an element for active, i.e., preventive safety, these VANETs can efficiently warn and inform drivers via direct wireless inter-vehicle communications. Thereby, the range of awareness of a driver is extended from current line-of-sight to the radio range of a wireless transceiver. With multi-hop communication, each vehicle can benefit from the locally sensed data of surrounding vehicles or from multi-hop access opportunities. Clearly, sensing, disseminating and retrieving information on the current surrounding shows a potential for improving transport efficiency and comfort.

Vehicular ad hoc networks represent a major technological challenge. Safety-related applications require a high degree of reliability and robustness of the communication system. Unfortunately, vehicular scenarios present adverse channel conditions due to the potential high mobility as well as density of nodes and high number of objects able to degrade the quality of a transmitted signal. All applications have to efficiently make use of the scarce bandwidth resource as well. Therefore, efficient and effective medium access control, routing, and data forwarding mechanisms play an essential role. Applications need clever data aggregation strategies to avoid an overload on the wireless medium.

Vehicular ad hoc networks represent a major business challenge. While the user does not have to pay for the use of a channel and data is stored and exchanged where it is created, a VANET will only work when enough vehicles are equipped with the corresponding technology. As a result, strong cost constraints on the VANET equipment exists as well as the need for attractive and convincing applications.

The FleetNet project investigated the feasability of vehicular ad hoc networks in the years 2000 to 2003, partly funded by the German Ministry of Education and Research (BMB+F). Leveraging the advances in wireless local area networks and positioning technologies, the project’s goal was to propose a communication system for vehicular ad hoc networks and to look into application and business aspects. Robert Bosch GmbH, DaimlerChrysler AG, NEC Europe Ltd., Siemens AG, Fraunhofer FOKUS, and the Universities of Braunschweig, Hamburg-Harburg, Hannover, Karlsruhe and Mannheim joined forces to achieve the task. In this book, major results and findings of the FleetNet project are brought together. We are aware of the fact that this is not the final word on vehicular ad hoc networks — 'shooting on a moving target' expresses well the current dynamics of this exciting field. The book is intended as a resource of information and of insights that have been collected while starting (almost) from scratch and ending with — to the best of our knowledge — the world's first real-world position-based vehicular ad hoc network.

The book covers aspects on physical, data link control, and network layer as well as on various application-specific issues. The chapters are ordered in a bottom-up approach with respect to communication protocol layering. Since each chapter is treating its topic in a self-contained way, the reader can choose his or her preference to find his or her appropriate order of reading through the book. Chapter 1 deals with the issue of time synchronization when a time-division multiple access method is used in a vehicular ad hoc network. Data link control aspects for FDMA, TDMA, and CDMA approaches are discussed in Chapter 2 where reservation-based medium access control is investigated in particular. How to forward emergency notification is the topic of Chapter 3. Chapters 4 and 5 deal with position-based routing and forwarding as seen from a conceptual point of view and as implemented in the FleetNet demonstrator, respectively. The integration of a vehicular ad hoc network with the Internet is discussed in Chapter 6. The remaining chapters 7, 8 and 9 all address VANET applications: several applications demonstrated with the FleetNet testbed, a self-organized traffic information system, and a location-based messaging application.

We like to thank all authors for their contributions, all partners for their support of the project, the BMB+F for funding, and the DLR, the handling project agency, particularly Andreas Kaatz for their constant support of the project. Hannes Hartenstein would also like to thank Moritz Killat for assisting in preparing this book. We hope that you, the reader, will find the book useful as a resource on ideas, concepts and achievements in the field of inter-vehicle communications based on ad hoc networking principles. In case you are interested to see how the story on vehicular ad hoc networks continues, please check the work done in the Car-to-Car Communication Consortium and in the NOW: Network on Wheels project (again partly funded by BMB+F).

Ulm, Karlsruhe, and Düsseldorf,
June 2005

Walter Franz
Hannes Hartenstein
Martin Mauve

Contents

1

Time Synchronization in Highly Dynamic Ad Hoc Networks

André Ebner[1], Lars Wischhof[2], Hermann Rohling[2],
Rüdiger Halfmann[3], and Matthias Lott[3]

[1] Hamburg University of Technology, Dept. of Telecommunications,
now with Audi Electronics Venture GmbH, Vorentwicklung Elektronik,
85045 Ingolstadt, Germany
andre.ebner@audi.de

[2] Hamburg University of Technology, Dept. of Telecommunications,
Eissendorfer Str. 40, 21073 Hamburg, Germany
{l.wischhof|rohling}@tuhh.de

[3] Siemens AG, Communications,
Sankt-Martin-Straße 76, 81541 München, Germany
{ruediger.halfmann|matthias.lott}@siemens.com

Summary. Active safety and advanced driver assistance systems based on vehicular ad hoc networks can significantly increase passenger safety and comfort. Within the FleetNet Project, a communication platform for vehicular ad hoc networks has been developed. As one potential basis for the air interface, the framework of the UMTS Terrestrial Radio Access Time Division Duplex (UTRA TDD) standard was investigated as a promising candidate. The multiple access scheme of UTRA TDD is based on a time division multiple access scheme which provides transmit resources in a framed and slotted structure. Consequently, time synchronization among the rapidly moving vehicles is required.

For time synchronisation purpose, two different options are described and analyzed: in the first approach, it is suggested to beneficially exploit the existence of a globally known time information coming from the Global Positioning System (GPS) for time synchronization. The second approach is based on a mutual decentralized synchronization scheme that works in a completely self-organizing fashion. In this case, the frame and slot timing of nodes are adjusted mutually.

Analytical as well as simulation results show that for slotted TDMA systems like UTRA TDD, time synchronization can be achieved without ground infrastructure even in highly dynamic environments.

1.1 Introduction

Ad hoc networks provide data communication services without being bound to an existing infrastructure. Within the last years, the application of ad hoc networks for a direct communication between vehicles (also known as car-to-car communication, C2CC) has aroused increasing interest. The main focus of C2CC can be seen in the area of active safety and advanced driver assistance systems, where typical applications range from emergency notification in cases of accidents, to distribution of Decentralized Floating Car Data (DFCD) for a Self-Organizing Traffic Information System (SOTIS) [1, 2]. Furthermore, typical established Internet applications like web browsing, e-mail or chat applications can be provided without a cellular wireless network infrastructure [3].

For this type of ad hoc networks, medium access schemes have been developed that rely on a slotted TDMA structure. One example is the proposed ad hoc extension of the Time Division Duplex mode of UMTS Terrestrial Radio Access (UTRA TDD) [4]. Consequently, time synchronization is required to align stations to the commonly used slot and frame structure.

One possibility to achieve time synchronization in a Vehicular Ad hoc NETwork (VANET) is to beneficially exploit the existence of globally known time information acquired from the Global Positioning System (GPS). In this case, all nodes within the network have access to a time reference with an error of significantly below 1 μs [5]. If under certain conditions, the reception from GPS satellites is disrupted, timing accuracy basically depends on the stability characteristics of the local oscillator. To estimate the duration that a VANET node can maintain time synchronization after disruption of GPS reception, the behavior of the local oscillator is analyzed in Sec. 1.4.

As an alternative to time synchronization based on GPS, a decentralized synchronization scheme will be proposed that works in a completely self-organizing fashion without being bound to a time reference coming from GPS. In this case, the frame and slot timing of nodes is adjusted mutually. The details of this approach are presented in Sec. 1.5 of this document.

1.2 Related Work

1.2.1 Inter-Base-Station Synchronization

In *cellular* TDMA systems, base stations provide a time and frequency reference for all associated mobile stations. Consequently, there is a common reference between all mobile nodes, associated to the respective base station. But beside synchronization of mobile stations, synchronous cellular systems like UTRA TDD require base stations to be time synchronized in order to fully exploit the system capacity [6]. Presently, the most commonly used method of time distribution is the utilization of GPS receivers, as described in Sec. 1.4. To be independent from GPS signals, distributed decentralized schemes for inter-base-station synchronization of *cellular* TDMA systems have been proposed:

An over-the-air autonomous technique was introduced by Akaiwa in 1991 [7]. There, all base stations monitor timing and power levels of surrounding stations and adapt their own timing according to a power-weighted sum of timing differences with respect to their own timing, multiplied by a fixed weighting factor to ensure stability. Eventually, this mechanism converges to a time-synchronized network. Nevertheless, one effect of this method is that a remaining constant timings drift of all base stations can be observed if the propagation delay between them cannot be accurately estimated and removed. This effect will be investigated in Sec. 1.5.

To reduce convergence time and to avoid timing drift, a modified version of this algorithm that does not require monitoring of the received power levels has been proposed by Chuang in 1994 [8]. There, a hierarchical value for each base station representing its respective synchronization quality is introduced. All satellite-based master base stations are given the lowest hierarchical value of zero. If bursts with hierarchy values lower than the own value are received, stations adapt their own timing. For this purpose, the mean observed timing offset to stations with the lowest hierarchy value is used for adaptation. Further, the hierarchy value of a receiving node is set to the lowest received value increased by one. This hierarchical distributed process is used to favor the propagation of satellite-based master timing within the network and therefore to avoid a constant clock drift. However, in the absence of any master timing station, a clock drift of nodes will be observed.

Another approach to compensate the constant clock drift was presented by Hirukawa in 1994 [9]. There, it is proposed to select only the closest stations (received power above a predefined level) for a mutual timing adaptation. This procedure is used to ensure that the propagation delay is small and therefore, only a low steady-state timing drift will be observed. Additionally, a quantization of the timing measurement is introduced: If the observed timing offset is smaller than the quantization threshold, it will be quantized to zero. If the propagation delays are smaller than the quantization threshold, a timing drift can be avoided. The synchronization accuracy is, however, rather low [9].

1.2.2 Synchronization for Ad Hoc Networks

The common assumption of the decentralized inter-base-station synchronization schemes is that the positions of participating stations are more or less static. Hence, only the convergence behavior of the respective mechanism to a final stable state is analyzed. In contrast, the network structure of an ad hoc network is subject to frequent topology changes. Particularly in traffic environments with large relative velocities, groups of nodes are approaching, forming new groups and reconfiguring the network topology. In this case, a dynamic performance analysis of the decentralization is of major importance.

For decentralized synchronization of ad hoc networks and especially for IVC, only few approaches have been published so far. In 1995, Akazawa adopted Akaiwa's inter-base-station synchronization scheme for a short-range

Inter-Vehicle Communication Network (ICWN) [10]. There, an adaptive weighting factor is introduced to improve the convergence of the system. Additionally, a protocol for nodes that enter an existing network (”new subscriber”) is proposed: New subscribers measure and adapt their own timing with the established network according to Akaiwa’s scheme but omit to transmit own signals until the difference between own timing and the timing of the network is below a certain threshold. Therefore, a single new subscriber can be merged. But it is still unclear how multiple groups of synchronized groups can be merged.

Another modified version of Akaiwa’s method for IVC is presented by Sourour and Nakagawa in [11]. Here, each vehicle transmits a periodic train of pulses using a devoted carrier frequency. Each node measures the power of pulses of other vehicles and shifts its own transmission towards a weighted average of other transmissions. The performance of this mechanism is measured in terms of time deviation, compared to a common virtual global timing, defined by the average slot timing of all nodes within the network.

A completely different approach is described by Hübner and Hoff in [12]: In this case, the number of surrounding nodes that are synchronous to a station is counted as a quality measure. This value is transmitted within each data packet. If a data packet with a higher quality measure is received, the timing of this station is adopted.

1.3 Time Synchronization of UTRA TDD Ad Hoc

The philosophy of FleetNet is to exploit an existing air-interface and incorporate slight modifications to come up in a very short time with a cost-effective solution that benefits from a mass market. This basic radio system together with the FleetNet network layer protocols enables the operation of the vehicular ad hoc network. The most obvious solution for the basic radio system would be to re-use an existing cellular system. However, time crucial key applications, e.g. driver assistance applications require a peer to peer communication. Particularly, safety applications should not rely on the availability of a cellular infrastructure. Furthermore, the amount of data to be exchanged between vehicles would congest a centralized network in heavy traffic conditions. Last not least it must be doubted that car holders and drivers would accept any business model which includes charges for data transmission.

The air-interface meeting the FleetNet requirements best is the UMTS Terrestrial Radio Access Time Division Duplex (UTRA TDD) standard [4]. One argument is the availability of an unlicensed frequency band at 2010 - 2020 MHz in Europe. Others are that UTRA TDD offers high flexibility with respect to asymmetric data flows and granularity for data transmission because of its code division multiple access (CDMA) component. Furthermore, it supports QoS since it is possible to reserve transmit capacity owing to its frame and slot structure. In addition, it allows communication over large distances and supports high velocities. In contrast to systems based on a WLAN standard, UTRA TDD was designed for a multipath propagation environment.

But since UTRA TDD was initially developed for operation in a cellular network structure, some modifications are required for an ad hoc operation. Besides the changes to Medium Access Control (MAC) and Radio Resource Management (RRM), also some modifications of the PHY layer are required to allow for an ad hoc operation. UTRA TDD comprises a Low Chip Rate (LCR, 1.28 Mchip/s) and a High Chip Rate (HCR, 3.84 Mchip/s) option. Since the LCR option is close to mass-market introduction, it was chosen for the FleetNet air-interface. However, the approach for the integration of an ad-hoc mode can easily be adapted to HCR.

The ad hoc mode of UTRA TDD uses the same time slotted radio frame structure as defined in the UTRA TDD specifications [13]. It is therefore required to apply a certain timing for the transmissions of mobile stations to avoid an overlap of the respective user bursts and to align all stations to the commonly used frame structure. Consequently, time synchronization of the mobile nodes is required. Within this work, two different approaches for time synchronization of UTRA TDD ad hoc will be described and analyzed:

1. **GPS-based time synchronization:** In the first approach, it is suggested to beneficially exploit the existence of a globally known time information coming from the Global Positioning System (GPS) for time synchronization.
2. **Decentralized time synchronization:** The second approach is based on a mutual decentralized synchronization scheme that works completely self-organizing without being bound to a time reference coming from GPS. In this case, the frame and slot timing of nodes is adjusted, mutually.

In the following section, the standard synchronization procedures as used in the cellular mode of UTRA TDD for the 1.28 Mcps option will be described. Subsequently, the differences and challenges for synchronization in a highly dynamic ad hoc network are explained, followed by the performance requirements for time synchronization of the UTRA TDD ad hoc mode.

1.3.1 Time Synchronization of UTRA TDD Cellular Mode

In the 1.28 Mcps option of UTRA TDD (LCR), time- and frequency synchronization can be derived from the Downlink Pilot Time Slot (DwPTS). The DwPTS also carries information that identifies the code group which is used by the respective cell.

The position of the DwPTS within a radio frame is depicted in Fig. 1.1 above. For cell search, a 4-step cell search procedure is used [14]:

Step 1. Search for DwPTS: During the first step of the initial cell search procedure, the UE uses the SYNC_DL (in DwPTS) to acquire DwPTS synchronization to a cell. This is typically done with one or more matched

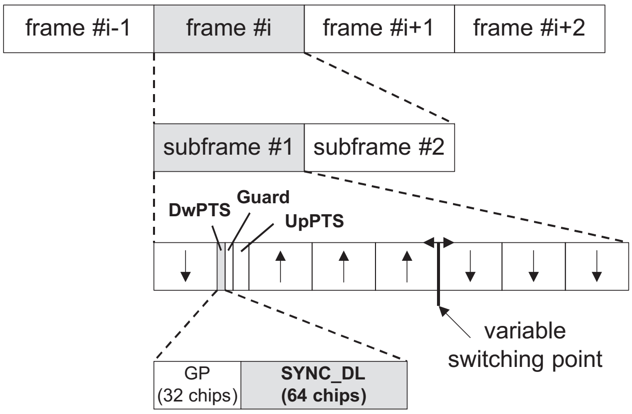


Fig. 1.1. Structure and position of DwPTS

filters (or any similar device) matched to the received SYNC_DL which is chosen from PN sequences set. During this procedure, the UE needs to identify which of the 32 possible SYNC_DL sequences is used.

Step 2. Scrambling and basic midamble code identification: During the second step of the initial cell search procedure, the UE receives the midamble of the Primary Common Control Physical Channel (P-CCPCH), the first time slot within the subframe. In the current low chip rate TDD option, each DwPTS code corresponds to a group of 4 different basic midamble codes. Therefore, the UE has to determine the used basic midamble code using a try and error technique. The same basic midamble code will be used throughout the frame. As each basic midamble code is associated with a scrambling code, the scrambling code is also known by that time. According to the result of the search for the right midamble code, UE may go to next step or go back to Step 1.

Step 3. Control multi-frame synchronization: During the third step of the initial cell search procedure, the UE searches for the head of multi-frame indicated by QPSK phase modulation of the DwPTS with respect to the P-CCPCH midamble. The control multi-frame is positioned by a sequence of QPSK symbols modulated on the DwPTS.

Step 4. Read the BCH: The (complete) broadcast information of the found cell in one or several BCHs is read. According to the result the UE may move back to previous steps or the initial cell search is finished.

1.3.2 Differences and Challenges in the UTRA TDD Ad Hoc Mode

In contrast to the standard cellular mode, the properties of an ad hoc operation lead to new challenges for the physical layer synchronization:

1. **Absence of base station:** In cellular wireless networks, the task of synchronization is organized in a centralized manner: Fixed base stations

provide a time and frequency reference for all associated mobile stations. Consequently, there is a common reference between all mobile nodes, associated to the respective base station. In an ad hoc network, there is no base station that could provide the reference parameters (as a master) for others. It is therefore recommended to develop new synchronization procedures that adapt to this situation.

2. **Frequent topology changes:** In an ad hoc network with frequent topology changes and particularly in traffic environments with large relative velocities, approaching groups of locally synchronized nodes are a major challenge: In order to avoid collisions, the approaching groups have to achieve at least a locally common timing without losing synchronization within their own group, respectively.
3. **No cell-specific parameters:** One important feature of the UTRA TDD ad hoc mode is the choice of cell parameters: In case of the standard cellular mode, UTRA TDD uses differing cell parameters for adjacent cells in order to reduce Inter Cell Interference (ICI), e.g. different scrambling codes, channelization codes and basic midamble codes. It is task of the cell search procedure to uniquely identify the parameter used in the cell where the respective UE is located. In case of an ad hoc network, there is no static cellular network structure and it is therefore not feasible to use differing parameters for adjacent cells.

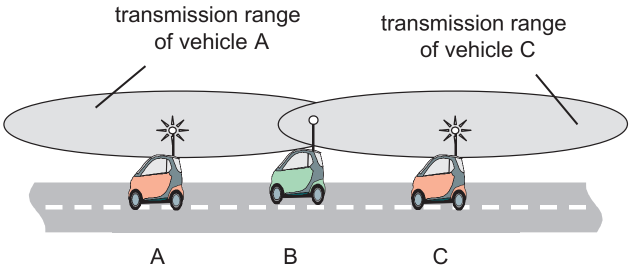

transmission range of vehicle A
transmission range of vehicle C
A
B
C

Fig. 1.2. Example for an ad hoc network structure

See Fig. 1.2 for an example situation: Vehicle B is located within the transmission ranges of vehicle A and vehicle C, respectively. If differing cell parameters for Vehicle A and C are assumed, it is not possible for Vehicle B to communicate to both mobile stations, simultaneously.

The only solution for vehicle B is to switch between the cell parameters of Vehicle A and Vehicle C, alternately. In a more complex environment, this would result in an unpredictable fluctuation of cell parameters within the network. As a result, the ad hoc mode of UTRA TDD uses only one fixed set of cell parameters for all mobile stations. Particularly, only one training sequence (midamble) is used by all ad hoc mobile stations. It is therefore not necessary to identify these parameters during the synchronization procedure.

1.3.3 Performance Requirements

To determine the required accuracy for time synchronization of UTRA TDD ad hoc, a transmission from node X_j to a second node X_i is shown in Fig. 1.3:

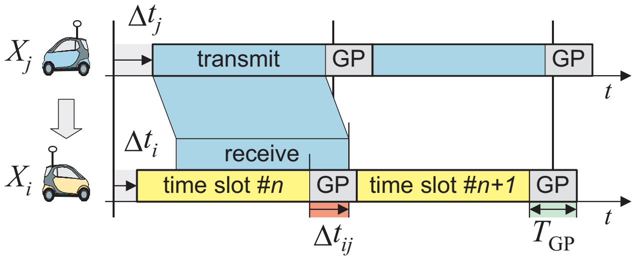


Fig. 1.3. Observed timing offset Δt_{ij}, observed at node X_i

Node X_i and X_j have an absolute time offset of Δt_i and Δt_j, compared to a virtual global time reference, respectively. An observed time offset Δt_{ij} of a burst, transmitted by X_j and received at node X_i can be calculated by:

$$\Delta t_{ij} = \Delta t_j - \Delta t_i + \frac{d_{ij}}{c}\,, \tag{1.1}$$

where d_{ij} denotes the distance between both nodes and c is the speed of light. To ensure at least a reception of bursts without overlap, the absolute value of the observed time offset Δt_{ij} has to be smaller than the guard period T_{GP}:

$$\Delta t_{ij} \overset{!}{<} T_{\text{GP}}\,. \tag{1.2}$$

Mobile nodes that fulfill at least the timing requirement in Eqn. (1.2) are called time synchronized, in the following.

The main difference between the GPS-based and the decentralized solution is that in the first approach, a common slot and frame timing of all nodes within the network is provided. In contrast, the objective of the second approach is to achieve only a locally common slot timing of nodes within their respective range of influence. As a result, the requirements of both solutions are different:

- For the **GPS-based synchronization scheme**, the absolute time error compared to a common reference is used as a performance criterion.
- For the **decentralized synchronization scheme**, only the relative time error compared to surrounding nodes is of importance (the observed time offset).

A detailed analysis of both approaches will be presented in the following sections 1.4 and 1.5.

1.4 GPS-Based Time Synchronization

A very common method of providing a reference signal for time synchronization is the utilization of GPS signals [15, 16]. The GPS-based synchronization approach is based upon the existence of a globally known time information coming from the Global Positioning System (GPS). Hence, the absolute time error compared to a common reference is used as a performance criterion.

1.4.1 Required Accuracy using GPS

In order to estimate the required accuracy of the absolute time offset Δt_i of a node X_i, we analyze the worst case that the timing inaccuracies of node X_i and X_j superimpose. Therefore, the maximum allowed (absolute) time error of an arbitrary node X_i, compared to a virtual global time reference is given by:

$$\left|\Delta t_{i,\max}\right| = \frac{1}{2}\left(T_{\mathrm{GP}} - \frac{d_{ij,\max}}{c}\right) \tag{1.3}$$

The required absolute timing accuracy $|\Delta t_{i,\max}|$ is the upper bound for the performance of the coarse time synchronization and only depends on the duration of the guard interval and the maximum distance between two nodes within their mutual range of transmission. In [3], it has been pointed out that for the realization of FleetNet multihop communications it is desirable that the radio system at least provides a transmission range of 1000 m. For example, assuming a maximum transmission range of 1500 m for UTRA TDD ad hoc, the required coarse time synchronization performance is shown in Table 1.1:

Table 1.1. Required timing accuracy for GPS-based coarse time synchronization

Parameter		**1.28 Mcps option (LCR)**	**3.84 Mcps option (HCR)**
Guard interval [μs]	T_{GP}	12.5	25
Max. propagation delay [μs]	$d_{ij,\max}$	5	5
Required timing accuracy [μs]	$\Delta t_{i,\max}$	3.75	10

1.4.2 Attainable Accuracy using GPS

The GPS satellites are controlled and operated by the United States Department of Defense (USDOD). The constellation includes at least 24 satellites that orbit the earth at a height of $20,200$ km in six fixed planes inclined 55° from the equator. Each satellite carries either rubidium or cesium oscillators, or a combination of both. The oscillators on board of the satellites are steered from USDOD ground stations and are referenced to Coordinated Universal Time (UTC). GPS provides a highly reliable time reference source with an absolute time error *significantly* below 1 μs [15]. Therefore, the accuracy requirement in Table 1.1 can easily be met using GPS time reference.

During *normal mode* operation of the GPS timing receiver, its internal precision oscillator is phase-locked to the GPS signal by comparing the time difference between the actual GPS time and the time derived from the local oscillator. While being locked to GPS, the receiver tracks four or more satellites to determine accurately the geographic position of the antenna. Advanced GPS receivers employ learning algorithms that measure the characteristics of the internal oscillator to be able to predict and to compensate changes during *holdover mode*.

If under certain conditions, the reception from GPS satellites is disrupted, the local oscillator of a node switches from *normal mode* to the *holdover mode*. During the loss of the GPS reference, the receiver uses data learned about the oscillator (acquired during normal mode) to control its output frequency and phase. In this case, timing accuracy depends on the stability characteristics of the local oscillator. The holdover mode ends when the reception from the GPS satellite can be resumed. To estimate the duration that a node can maintain coarse time synchronization after switching to holdover mode, the receiver clock is modeled using a two-state model based on Allan variance parameters as described in [17]. The details of this model are presented in the following section.

1.4.3 Oscillator Clock Model

If the reception from GPS satellites is disrupted, the timing accuracy depends on the stability characteristics of the local oscillator. In this context, it is important to stress the difference between *frequency offset* and *stability* of an oscillator. The frequency offset is a measure of how well an oscillator produces its nameplate frequency, or how well an oscillator is adjusted. It does not tell us about the inherent quality of an oscillator [18]. Stability, on the other hand, indicates how well an oscillator can produce the same frequency over a given period of time.

For the following analysis, it is assumed that the receiver uses data learned about the oscillator (acquired during normal mode) to control its output frequency and phase. Particularly, the frequency and time offset at $t = 0$ is zero at the beginning of the holdover mode. Consequently, timing accuracy depends only on the stability characteristics of the local oscillator.

The behavior of real oscillators has been studied, extensively. By recommendation of the Institute of Electrical and Electronics Engineers (IEEE), the standard measurement for oscillator stability is the so-called Allan deviation, defined by:

$$\sigma_y\left(\tau\right) = \sqrt{\frac{1}{2\left(M-1\right)} \sum_{k=1}^{M-1} \Big[y(k+1) - y(k)\Big]^2}\,, \tag{1.4}$$

where $y(k)$ is a set of normalized frequency offset measurements, equally spaced and averaged in segments with a period duration of τ [19]:

$$y(k) = \frac{1}{\tau}\Big[\mathit{\Delta} t(k+1) - \mathit{\Delta} t(k)\Big]\,. \tag{1.5}$$

The integer value M is the number of time offset measurements $\mathit{\Delta} t(k)$. A typical Allan deviation plot is shown in Fig. 1.4 using a double-logarithmic scale.

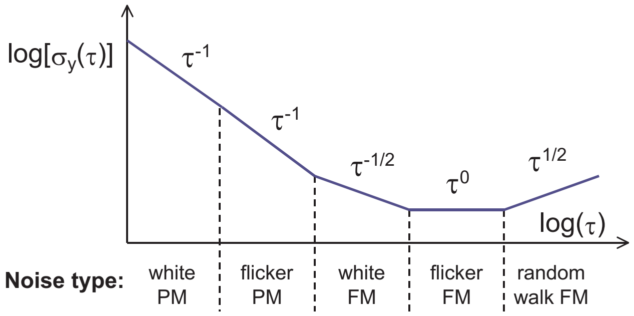


Fig. 1.4. Allan deviation plot showing different noise sources

Fig. 1.4 shows that the stability of the device improves as the averaging period τ gets longer, since some noise types can be removed by averaging. At some point, however, the oscillator will reach its flicker floor, and from a practical point of view, no further gains will be made by averaging. The flicker floor is the point where the white noise processes begin to be dominated by non-stationary processes such as random walk frequency modulation [18]. In Fig. 1.4 it can also be observed that there is a direct relationship between the slope of the Allan deviation and the type of noise process. For instance, the white frequency modulation noise has a slope of $\tau^{-1/2}$, the random walk frequency modulation has a slope of $\tau^{1/2}$.

In contrast to the time domain characterization using the Allan deviation, the properties of an oscillator can also be characterized in the frequency domain. For that purpose, the power spectral density $S_y(f)$ of the frequency deviations y is used as a common measurement. The spectral density is a measure of the power present at different Fourier frequencies. The Fourier frequency is the deviation from the nominal frequency. Therefore, $S_y(f)$ is associated with the power in the residuals (the fluctuations of the actual frequency around the nominal carrier frequency value). Similar to the Allan deviation, each noise process corresponds to a different slope of $S_y(f)$:

$$S_y\left(f\right) = \sum_{\alpha=-2}^{2} h_\alpha \cdot f^\alpha \,. \tag{1.6}$$

The relationship between Allan variances in time domain and power spectral densities in the frequency domain for the different noise processes has been verified experimentally [20] and by computation [21]. These h_α parameters (also called *Allan Variance Parameters*, [17]) are used in the two-state model of the oscillator behavior. It has been stated in [17] that the drift characteristics of a real oscillator can be approximately modelled by a superposition of the following two stochastic processes:

1. White Frequency Modulation (WFM)
2. Random Walk Frequency Modulation (RWFM)

This assumption leads to the (time-continuous) system shown in Fig. 1.5.

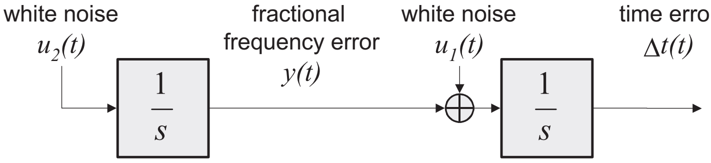


Fig. 1.5. Block diagram of two-state clock model

Mathematically, this system can be described by the following state-vector representation:

$$\begin{bmatrix} \dot{\Delta t}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta t(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \,, \tag{1.7}$$

where $\Delta t(t)$ is the resulting clock time error. The two inputs u_1 and u_2 are uncorrelated, zero-mean, Gaussian distributed white noise inputs and have standard deviations of σ_1 and σ_2, respectively. Using the method, proposed in [17], the input variances can be determined from the Allan variance parameters:

$$\begin{aligned} E\left\{u_1 \cdot u_1\right\} &= \sigma_1^2 = 2h_0 \\ E\left\{u_2 \cdot u_2\right\} &= \sigma_2^2 = 8\pi^2 h_{-2} \,. \\ E\left\{u_1 \cdot u_2\right\} &= 0 \end{aligned} \tag{1.8}$$

To simulate the system, it is required to find the corresponding discrete-time model to Eqn. (1.7). Using the algebraic manipulations described in [17], the discrete-time model is given by

$$\begin{bmatrix} \Delta t(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta t(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix} , \tag{1.9}$$

where w_1 and w_2 are white noise inputs with the covariance matrix $\mathbf{Q}$, given by

$$\mathbf{Q} = E\left[\mathbf{w}\mathbf{w}^T\right] = \sigma_1^2 \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix} + \sigma_2^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} , \tag{1.10}$$

and T is the duration between two discrete measurements (”*sampling time*”). The time-dependent variance of the accumulated time error is given by

$$\sigma^2(T) = E\Big[\Big(\Delta t(T)\Big)^2\Big] = E\Big[w_1^2\Big] = \sigma_1^2 T + \sigma_2^2 \frac{T^3}{3} . \tag{1.11}$$

Consequently, the random walk frequency modulation noise process dominates the accumulated timing error for large values of T.

1.4.4 Performance Analysis

The range of different oscillator types includes simple uncompensated crystal oscillators (XO), temperature compensated (TCXO), microcomputer compensated (MCXO) and oven or double-oven controlled compensated crystal oscillators (OCXO). Additionally, as a member of the atomic oscillator family, Rubidium oscillators use the rubidium resonance frequency to control the frequency of a quartz oscillator.

Each oscillator type has advantages and disadvantages in terms of performance and price. Table 1.2 shows typical values of h_0, h_{-1} and h_{-2} for various types of oscillators widely used in GPS receivers.

Table 1.2. Typical Allan variance parameters for various timing standards [17]

Oscillator type	h_0	h_{-1}	h_{-2}
Crystal Quartz	$2 \cdot 10^{-19}$	$7 \cdot 10^{-21}$	$2 \cdot 10^{-20}$
Ovenized Crystal	$8 \cdot 10^{-20}$	$2 \cdot 10^{-21}$	$4 \cdot 10^{-23}$
Rubidium	$2 \cdot 10^{-20}$	$7 \cdot 10^{-24}$	$4 \cdot 10^{-29}$

Fig. 1.6 shows the Allan deviation plots using the parameters shown in Table 1.2. It can be observed that the crystal oscillator reaches its flicker floor at a value of approximately one second. From this point, the non-stationary

random walk frequency modulation processes becomes dominant. In contrast, the ovenized crystal shows a much lower flicker floor which is reached at a time of 10 seconds. The rubidium clock shows the best performance in terms of stability.

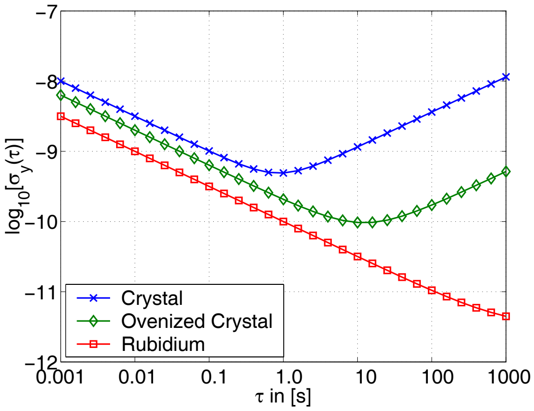
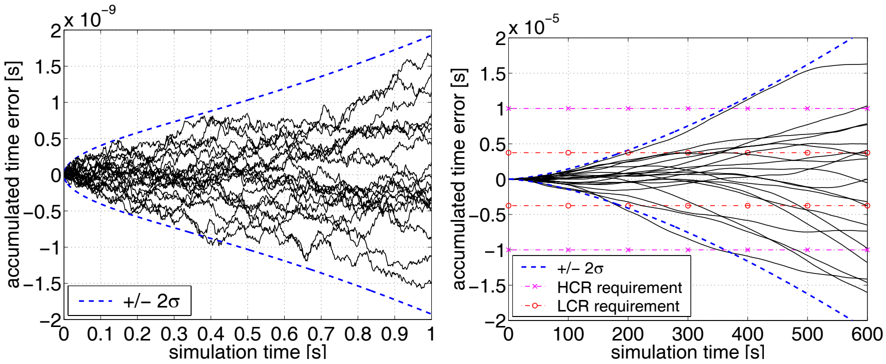


Fig. 1.6. Typical Allan deviation plots for various timing standards

Using the assumptions described above, computer simulations have been carried out to estimate the duration that a receiver can maintain coarse time synchronization during the Holdover mode. Fig. 1.7 shows the accumulated time error for 20 simulated crystal oscillator based clocks and the interval of +/- 2 s, where 95% of all clock errors are located. Additionally, the required timing accuracy for coarse time synchronization of the HCR and LCR option are depicted (see also Table 1.1).

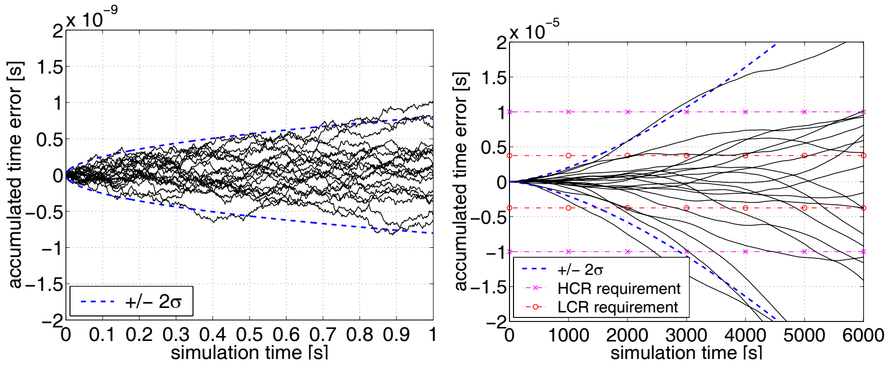


(a) Small time scale

(b) Large time scale

Fig. 1.7. Accumulated timing error for crystal oscillator

Like depicted in Fig. 1.7, using the crystal oscillator, coarse time synchronization can be maintained in situations without GPS reception for 350 s in 95% of the cases in case of UTRA TDD HCR. For the LCR option, coarse time synchronization can be maintained for approx. 180 s in 95% of all cases.



(a) Small time scale (b) Large time scale

Fig. 1.8. Accumulated time error for ovenized crystal oscillator

Fig. 1.8 shows the simulated time error for the ovenized crystal oscillator. Coarse time synchronization can be maintained in holdover mode almost 3000 s in 95% of the cases in case of UTRA TDD HCR. For the LCR option, coarse time synchronization can be maintained for approx. 1500 s in 95% of all cases.

1.5 Decentralized Time Synchronization

Within the last sections, a GPS-based approach for coarse time synchronization of the FleetNet air-interface has been presented. In contrast, the second approach presented in the following is based on a mutual decentralized synchronization scheme. In this case, the coarse time synchronization (frame and slot timing of nodes) is performed completely self-organizing without being bound to a GPS time reference.

The basic idea of the proposed decentralized synchronization scheme is to achieve a locally common slot and frame timing by a mutual adaptation of the individual node timing. As described in Sec. 1.2, one of the first algorithms to realize a decentralized time synchronization was proposed by Akaiwa in 1991 for an autonomous inter-base-station synchronization [7]. There, each station monitors the timing and power levels of surrounding stations and adapts its own timing according to a power-weighted sum of observed timing differences with respect to its own timing. Several modified versions of Akaiwa's scheme can be found in the literature [8, 10, 11, 22, 23]. All these schemes have in common that time synchronization is achieved by shifting the timing of each node towards a weighted average of the surrounding node timings. This mechanism will be called the ”standard method”, in the following. One drawback of the standard method is that caused by the propagation delay, a systematic timing drift and a remaining timing offset between nodes can be observed, in a steady state. Depending on the network topology, this timing offset can be large enough to violate the constraints for a synchronous operation.

In order to avoid the systematic timing drift and the large remaining timing offset of the standard method, the decentralized synchronization scheme proposed in the following section uses a modified version of the mutual timing adaptation scheme [24]. Using the proposed scheme, a systematic drift of the node timing can be avoided, completely. Compared to existing schemes, the remaining timing offset is much smaller and the requirements for synchronous operation can be met.

Another important feature of the proposed synchronization algorithm is the *increased synchronization range* compared to regular data transmission ranges: In the literature, timing offsets to surrounding stations are measured using regular data demodulation. Consequently, the synchronization process is performed when the respective nodes are already in a mutual interference range. In contrast, our approach uses an increased synchronization range compared to regular data transmission. Therefore, synchronization of approaching groups is performed even before the mutual interference becomes severe.

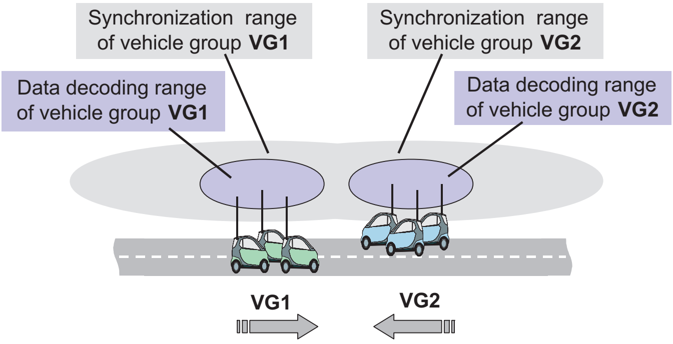

Synchronization range of vehicle group **VG1**
Synchronization range of vehicle group **VG2**
Data decoding range of vehicle group **VG1**
Data decoding range of vehicle group **VG2**
VG1
VG2

Fig. 1.9. Synchronization range and data decoding range

This fact is depicted in Fig. 1.9: Two groups of locally synchronized vehicles VG1 and VG2 are heading towards each other. Before nodes within VG1 are able to detect data from nodes of VG2, the synchronization ranges of both groups partly overlap and the synchronization process adjusts the slot and frame timing of the respective nodes.

1.5.1 Proposed Synchronization Algorithm

The synchronization procedure consists of two steps: First, the timing of a received burst is acquired, and in a second step, the own timing is adapted according to the observed time difference to the node that transmitted the respective burst.

Slot Timing Acquisition

For one-shot synchronization of received bursts, a correlation sequence $m[k]$ of length K is transmitted within each time slot. The sequence is chosen to have noise-like autocorrelation properties and can be placed either at the beginning (as *preamble*) or in the middle of a burst (as *midamble*).

During every time slot without own transmission, each node i correlates its received signal $r[n]$ with the correlation sequence $m[n]$:

$$\varPsi_n = \sum_{k=0}^{K-1} r[n+k] \cdot m^*[k] \,. \tag{1.12}$$

By finding the argument n that maximizes the absolute value of $\varPsi_n$, the observed timing offset $\varDelta t_{ij}$ at node i to an arbitrary node j that transmitted during the current time slot can be estimated by

$$\hat{\varDelta t}_{ij} = T_{\text{sample}} \cdot \left(\arg\max_n |\varPsi_n| - n_0 \right) \,, \tag{1.13}$$

where n_0 is the fixed position of the correlation sequence within the time slot and T_{sample} is the sampling time. As shown in Fig. 1.10, the observed timing offset $\varDelta t_{ij}$ includes the actual time difference between both stations and the propagation delay between them:

$$\varDelta t_{ij} = \varDelta t_j - \varDelta t_i + \frac{d_{ij}}{c} \,, \tag{1.14}$$

where c denotes the speed of light and d_{ij} denotes the distance between node i and j.

The geographic range where a slot timing acquisition is possible is denoted as the synchronization range R_{sync}. By choosing an appropriate length K of the correlation sequence, the synchronization range can be adjusted. An important fact of the decentralized synchronization is that R_{sync} is much larger than the range of possible data decoding. As a result, approaching vehicle groups are able to mutually synchronize before both groups merge for communication [22].

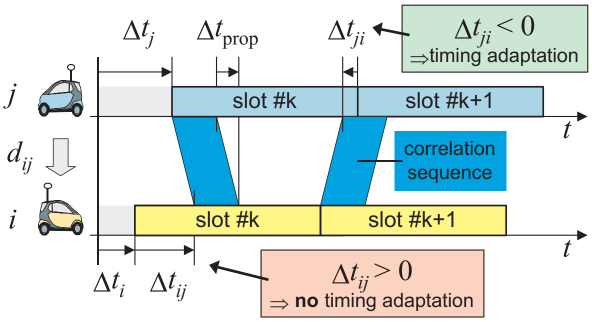


Fig. 1.10. Basic principle of the new proposed mechanism for decentralized time synchronization

Slot Timing Adaptation

At the end of a successful timing acquisition phase, node i adapts its own slot timing Δt_i according to

$$\Delta t_{i,\text{new}} = \begin{cases} \Delta t_{i,\text{old}} + w \cdot \hat{\Delta t}_{ij} & \text{if } \hat{\Delta t}_{ij} < 0, \\ \Delta t_{i,\text{old}} & \text{otherwise.} \end{cases} \tag{1.15}$$

where the parameter w denotes a weighting factor that determines the stability and convergence behavior of the algorithm.

The most important fact of the new proposed synchronization scheme is that according to Eqn. (1.15), only negative observed time offsets Δt_{ij} are compensated. The reason to ignore positive offsets is that with each acquired time offset, not only the actual difference of the node timings is acquired but also the distance-dependant propagation delay, see Eqn. (1.14). If positive acquired offsets would be compensated, a systematic timing drift would be observed, since each node tries to compensate the measured propagation delay. By compensating only negative time offsets in the new proposed mechanism, this systematic drift of the node timings is prevented.

An example is shown in Fig. 1.10: In the first slot, a positive time offset Δt_{ij} is acquired at node i and therefore no timing adaptation is initiated. In the second slot, node j acquires a negative time offset, which will be compensated. If two nodes i and j are within respective synchronization range, it will *never* occur that more than one of them will acquire a negative time offset to the other one, respectively. As a result, from a pair of nodes only one of them adapts its timing until the offset is compensated.

1.5.2 System Model

For the analysis, the network of N nodes is regarded as an *undirected graph*, where there is an edge between node i and j, if they are within synchronization range, respectively. The topology of the graph can be represented by the *adjacency matrix* **A**, defined as follows:

$$\mathbf{A} = \left\{a_{ij}\right\}, \, a_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq R_{\text{sync}} \wedge i \neq j, \\ 0 & \text{otherwise} \end{cases} \tag{1.16}$$

The *degree matrix* **N** is a diagonal matrix

$$\mathbf{N} = \operatorname{diag}\left[N_1, \ldots, N_N\right], \tag{1.17}$$

where $N_i = \sum_j a_{ij}$ denotes the number of nodes within synchronization range of node i.

The matrix **H** is introduced, which contains the propagation delay between two nodes i and j under the condition that they are within their respective synchronization range:

$$\mathbf{H} = \left\{h_{ij}\right\}, \text{where } h_{ij} = \frac{d_{ij} \cdot a_{ij}}{c}. \tag{1.18}$$

For the following analysis, the continuous time t is divided into fixed intervals of length T. The whole system is analyzed at discrete points in time $t = kT$ with $k \in \mathbb{N}_0$. For simplicity, it is assumed that within an interval T, an arbitrary node i is able to acquire the observed timing offset of *all* neighbors, respectively. Accordingly, a timing update of node i from time k to $k+1$ can be expressed by

$$\Delta t_i(k+1) = \Delta t_i(k) + w \sum_{j=1}^{N} a'_{ij}(k) \Delta t_{ij}(k), \tag{1.19}$$

where $a'_{ij}(k) = a_{ij}$ for the standard method and

$$a'_{ij}(k) = \begin{cases} a'_{ij} & \text{if } \Delta t_{ij}(k) < 0, \\ 0 & \text{otherwise}, \end{cases} \tag{1.20}$$

for the new proposed mechanism.

1.5.3 Performance Analysis

In the following, the steady-state-behavior of the standard and the proposed new mechanism is evaluated analytically, similar to a work in [23]. In contrast to [23], where a *complete graph*[4] is assumed, the graph analyzed in this paper is assumed to be connected, but non-complete.

[4] In a **complete graph**, there is an edge between all possible pairs of nodes, respectively.

Analysis of Standard Mechanism

By inserting Eqn. (1.14) in Eqn. (1.19), a single timing update for the standard method can be expressed as follows:

$$\begin{aligned}\varDelta t_i(k+1) = \varDelta t_i(k) + w \sum_{j=1}^{N} a_{ij} \varDelta t_j(k) \\ -w \sum_{j=1}^{N} a_{ij} \varDelta t_i(k) + \frac{w}{c} \sum_{j=1}^{N} a_{ij} d_{ij} \ . \end{aligned} \tag{1.21}$$

For simplicity, a vector notation is used in the following. Therefore, the timing of all nodes at time k is represented by a single vector:

$$\mathbf{\Delta t}(k) = [\varDelta t_1(k)\,,\, \varDelta t_2(k)\,,\, \cdots\,,\, \varDelta t_N(k)]^{\mathrm{T}} \ . \tag{1.22}$$

Consequently, a timing update can expressed in vector notation by

$$\mathbf{\Delta t}(k+1) = \mathbf{M}\mathbf{\Delta t}(k) + \mathbf{b}\,, \tag{1.23}$$

where

$$\mathbf{M} = \begin{bmatrix} 1-N_1 w & a_{12}w & \cdots & a_{1N}w \\ a_{12}w & 1-N_2 w & \cdots & a_{2N}w \\ \vdots & \vdots & \ddots & \vdots \\ a_{1N}w & a_{2N}w & \cdots & 1-N_N w \end{bmatrix} \tag{1.24}$$

$$\mathbf{b} = \frac{w}{c}\Big[\sum_{j=1}^{N} a_{1j}\, d_{1j}\,,\, \cdots\,,\, \sum_{j=1}^{N} a_{Nj}\, d_{Nj}\Big]^{\mathrm{T}} \ . \tag{1.25}$$

Using the definition of matrix $\mathbf{A}$ and $\mathbf{N}$, $\mathbf{M}$ can be written as

$$\mathbf{M} = \mathbf{I} - w\left(\mathbf{N} - \mathbf{A}\right)\,, \tag{1.26}$$

where $\mathbf{I}$ is a $N \times N$ unity matrix. With the definition of $\mathbf{H}$ in Eqn. (1.18), the vector $\mathbf{b}$ becomes

$$\mathbf{b} = w\,\mathbf{H}\left[1,\ldots,1\right]^{\mathrm{T}} \ . \tag{1.27}$$

Since $\mathbf{M}$ is symmetric, $\mathbf{M}$ can be diagonalized:

$$\mathbf{M} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}}\,, \tag{1.28}$$

where $\mathbf{Q} = [\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_N}]$ is a matrix containing orthonormal eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of the corresponding eigenvalues λ_m . Particularly, the eigenvalue $\lambda_1 = 1$ and the corresponding eigenvector

$$\mathbf{q}_1 = \frac{1}{\sqrt{N}}\left[1,\ldots,1\right]^{\mathrm{T}} \tag{1.29}$$

exist [23]. In order to reach a stable synchronization, it is required that for all other eigenvalues λ_m, the following condition holds:

$$\|\lambda_m\| < 1\ ,\ m = 2, \ldots, N\,. \tag{1.30}$$

The eigenvalue decomposition of $\mathbf{M}$ is now used in the update formula Eqn. (1.23):

$$\boldsymbol{\Delta}\mathbf{t}(k+1) = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Delta}\mathbf{t}(k) + \mathbf{b}\,. \tag{1.31}$$

Assuming a vector $\boldsymbol{\Delta}\mathbf{t}(0)$ of the initial timing states, the recursive update formula can be written as

$$\boldsymbol{\Delta}\mathbf{t}(k+1) = \mathbf{Q}\boldsymbol{\Lambda}^{k+1}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Delta}\mathbf{t}(0) + \sum_{n=0}^{k}\mathbf{Q}\boldsymbol{\Lambda}^{n}\mathbf{Q}^{\mathrm{T}}\mathbf{b}\,. \tag{1.32}$$

Timing drift in a steady state

In the following, the difference vector $\mathbf{d}(k)$ of two consecutive timing vectors is considered.

$$\mathbf{d}(k) = \boldsymbol{\Delta}\mathbf{t}(k+1) - \boldsymbol{\Delta}\mathbf{t}(k) \tag{1.33}$$

Using Eqn. (1.32) and the fact that $\boldsymbol{\Lambda}$ is a diagonal matrix, $\mathbf{d}(k)$ can be expressed by

$$\begin{aligned}\mathbf{d}(k) = &\left(\sum_{m=1}^{N}\lambda_m^k\,(\lambda_m - 1)\,\mathbf{q}_m\mathbf{q}_m^{\mathrm{T}}\right)\boldsymbol{\Delta}\mathbf{t}(0)\\ &+ \left(\sum_{m=1}^{N}\lambda_m^k\mathbf{q}_m\mathbf{q}_m^{\mathrm{T}}\right)\mathbf{b}\,.\end{aligned} \tag{1.34}$$

Under the assumption that the condition in Eqn. (1.30) is fulfilled, the influence of all eigenvalues except $\lambda_1 = 1$ disappears for large values of k:

$$\lim_{k\to\infty}\lambda_m^k = 0\ \forall\ m \neq 1\ , \tag{1.35}$$

therefore, the constant drift vector $\mathbf{d}$ in steady state can be calculated by

$$\mathbf{d} = \lim_{k\to\infty}\mathbf{d}(k) = \mathbf{q}_1\,\mathbf{q}_1^{\mathrm{T}}\,\mathbf{b}\,. \tag{1.36}$$

Using Eqn. (1.29) and the definition of $\mathbf{b}$ in Eqn. (1.27), the drift vector $\mathbf{d}$ can be expressed by

$$\mathbf{d} = \frac{w}{N}\left[1,\ldots,1\right]^{\mathrm{T}}\left[1,\ldots,1\right]\mathbf{H}\left[1,\ldots,1\right]^{\mathrm{T}}\,. \tag{1.37}$$

The product of the right three factors is a scalar value $\tilde{H}$, which can be calculated by summing all elements of matrix $\mathbf{H}$. Therefore, $\mathbf{d}$ becomes

$$\mathbf{d} = w\,\frac{\tilde{H}}{N}\left[1,\ldots,1\right]^{\mathrm{T}}\,. \tag{1.38}$$

Using the standard mechanism, all nodes of the network drift by the same amount per time interval, which is the weighted sum of all propagation delays, in a steady state.

Remaining timing offset in a steady state

Using the result from Eqn. (1.38) for the constant drift vector $\mathbf{d}$, the remaining timing offset $\mathbf{\Delta t}$ of nodes in a steady state can be calculated.

By combining Eqn. (1.23) and the definition of the drift vector in Eqn. (1.33), the following equation holds:

$$\mathbf{M}\mathbf{\Delta t}(k) - \mathbf{\Delta t}(k) + \mathbf{b} = \mathbf{d}(k)\ . \tag{1.39}$$

For a steady state, Eqn. (1.39) becomes

$$\left(\mathbf{M} - \mathbf{I}\right)\mathbf{\Delta t} = \mathbf{d} - \mathbf{b}\,, \tag{1.40}$$

which represents a system of linear equations. Using Eqn. (1.26), Eqn. (1.27) and Eqn. (1.37), the vector equation in Eqn. (1.40) can be expressed by

$$\big(\mathbf{N} - \mathbf{A}\big)\mathbf{\Delta t} = \frac{1}{c}\left(\mathbf{H} - \frac{\tilde{H}}{N}\mathbf{I}\right)\big[1, \ldots, 1\big]^{\mathrm{T}}\,. \tag{1.41}$$

The left matrix $\mathbf{L} = \mathbf{N} - \mathbf{A}$ is known as the *Laplacian matrix* of an undirected graph. It has

$$\operatorname{rank}\big(\mathbf{N} - \mathbf{A}\big) = N - 1\,, \tag{1.42}$$

if the graph is connected [25]. Therefore, the system of equations in Eqn. (1.41) is underdetermined by one. By setting a single value, e.g. $\Delta t_1 = 0$, a solution for $\mathbf{\Delta t}$ can be found.

A very interesting result is that Eqn. (1.41) is completely independent from the weighting factor w. As a result, the remaining timing offset is only influenced by the network topology.

Analysis of Proposed Mechanism

In contrast to the standard timing method, the proposed new algorithm ignores nodes with positive values of the observed timing offset Δt_{ij}, see Eqn. (1.15). This means that compared to the standard method, not all nodes within synchronization range are considered for the timing adaptation of a node.

Particularly, if two nodes i and j are within respective synchronization range, it will never occur that more than one of them will acquire a negative time offset to the other one, respectively. Thus, from a pair of nodes only one of them adapts its timing until the offset is compensated. As depicted in Fig. 1.10, node j will adapt its timing until the following equation holds:

$$\Delta t_{ji} = 0 \Rightarrow \Delta t_{ij} = 2\,\frac{d_{ij}}{c}\ . \tag{1.43}$$

In a steady state, two nodes are kept synchronous within a time interval of twice the propagation delay between them. In this case, the factor $a'_{ij}(k)$ in Eqn. (1.19) is zero for each node j. Consequently,

Table 1.3. Scenario Parameters and Analytical Results for the Standard Method

Node i	1	2	3	4	5	6	7	8	9	10
X-Position of node [m]	-40	-30	-20	-10	0	1000	2000	3000	4000	5000
Nodes in sync. range	4	4	4	4	5	3	2	2	2	1
Average prop. delay [μs]	0.083	0.058	0.050	0.058	0.733	3.333	3.333	3.333	3.333	3.333
Steady-state-drift [$\mu s/s$]	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93	6.93
Steady-state-timing Δt_i [μs]	0	-0.02	-0.03	-0.02	3.2	15.87	25.33	31.6	34.67	34.53

$$\begin{aligned}&\boldsymbol{\Delta}\mathbf{t}(k+1) = \boldsymbol{\Delta}\mathbf{t}(k) \Rightarrow \mathbf{d}(k) = \mathbf{0}\ ,\\ &\text{if } |\Delta t_{ij}| \leq \frac{2\, d_{ij}}{c}\ ,\ i,j \in \{1,\ldots,N\}\ . \end{aligned} \tag{1.44}$$

In a steady state, there is no systematic drift of the timing like observed for the standard method.

1.5.4 Simulation Results

In order to validate the analytic results from Sec. 1.5.3, simulations have been carried out. The parameters for one example scenario are listed in Table 1.3. Additionally, the scenario is depicted in Fig. 1.11: Beginning from the left, five nodes with a distance spacing of 10 m and five nodes with spacing of 1000 m are placed on a line, respectively. A fixed synchronization range R_{sync} of 1000 m and a weighting factor $w = 0.02$ is assumed. The duration of an update interval is set to $T = 0.01$ s. The initial timing offset Δt_i of each node is a pseudo-random number taken from a uniform distribution in the range from zero to 100 μs.

The chosen scenario has two important properties: First, the graph of the network is non-complete, but connected; second, the average propagation delay to nodes within synchronization range differs from node to node, see Table 1.3.

Results for Standard Mechanism

The results for the simulated timing Δt_i are shown in

Fig. 12(a) for the standard mechanism. As derived analytically, all nodes within the network drift by the same amount per time interval, in steady

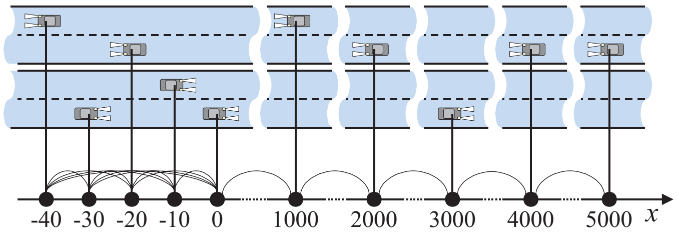
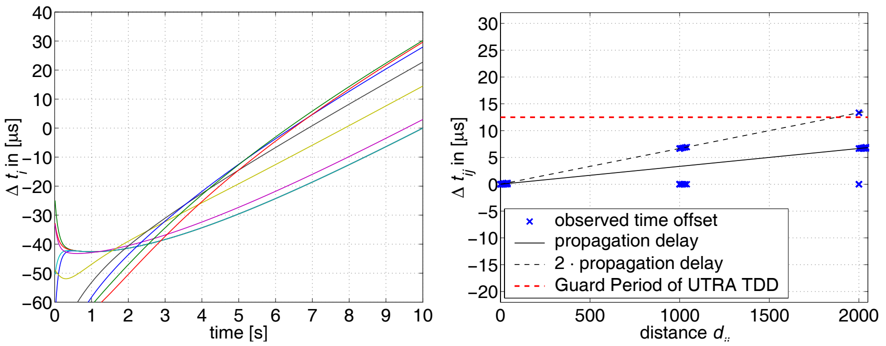


Fig. 1.11. Scenario with $N = 10$ nodes and a synchronization range of 1000 m. Pairs of nodes within respective synchronization range are connected by an arc.



(a) absolute timing Δt_i, depending on the simulation time

(b) observed timing offset Δt_{ij} in a steady state, depending on nodal distances

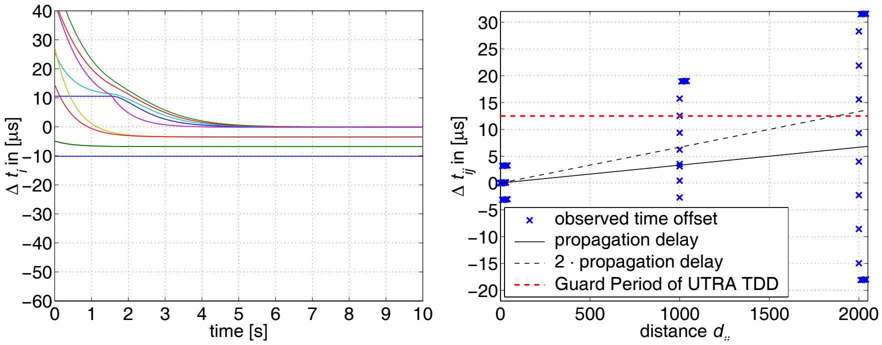
Fig. 1.12. Simulation results for the standard method; weighting $w = 0.02$, update interval $T = 0.01$ s

state. It can be observed that the timing of nodes does not converge to a common, single line: There is a remaining timing offset between the nodes, in a steady state. To be able to compare the simulated remaining timing offset with the analytical results in Table 1.3, the curves in Fig. 1.12 are shifted, so that $\Delta t_1 = 0$ at the end of the simulation. It can be observed that analytical values and simulation results match exactly.

The simulation results for the observed timing offset Δt_{ij} depending on the distance d_{ij} between pairs of nodes i and j are depicted in Fig. 12(a), for a steady state. It is shown that Δt_{ij} can be several times larger than the propagation delay between i and j. As a result, the timing constraints for a synchronous operation of a slotted system can be violated. For example, the Time Division Duplex mode of UMTS Terrestrial Radio Access (UTRA TDD), which is proposed as a possible air-interface for Vehicular Ad-hoc Networks (VANETs), foresees a guard period of 12.5μs [5]. As depicted in Fig. 12(b), this requirement is violated for nodes even with $d_{ij} \leq 1000$ m.

Results for Proposed Mechanism

For the proposed mechanism, the simulation results of the node timing Δt_i are shown in Fig. 13(a). It can be observed that in contrast to the standard mechanism, a systematic timing drift can be completely avoided.



(a) absolute timing Δt_i, depending on the simulation time

(b) observed timing offset Δt_{ij} in a steady state, depending on nodal distances

Fig. 1.13. Simulation results for the standard method; weighting $w = 0.02$, update interval $T = 0.01$ s

As depicted in Fig. 13(b), all nodes are kept synchronous within a time interval of twice the propagation delay between them. As an example, the timing constraints for a synchronous operation of UTRA TDD can be met for all nodes with $d_{ij} \leq 1800$ m.

1.6 Conclusions

In this chapter, the challenge of time synchronization in a highly dynamic ad hoc network has been discussed. Two different options are described and analyzed: in the first approach, it is suggested to beneficially exploit the existence of a globally known time information coming from the Global Positioning System (GPS) for time synchronization. The second approach is based on a mutual decentralized synchronization scheme that works in a completely self-organizing fashion. In this case, the frame and slot timing of nodes is adjusted mutually. Analytical as well as simulation results show that for slotted TDMA systems like UTRA TDD, time synchronization can be achieved without ground infrastructure even in highly dynamic environments.

References

1. Wischhof, L., Ebner, A., Rohling, H., Lott, M., Halfmann, R.: SOTIS - A Self-Organizing Traffic Information System. Proceedings of the 57th IEEE Vehicular Technology Conference (VTC '03 Spring), Jeju, Korea (2003)
2. Wischhof, L., Ebner, A., Rohling, H., Lott, M., Halfmann, R.: Adaptive Broadcast for Travel and Traffic Information Distribution Based on Inter-Vehicle Communication. Proceedings of the IEEE Intelligent Vehicles Symposium (IV '03), Columbus, Ohio, USA (2003)
3. Hartenstein, H., Bochow, B., Lott, M., Ebner, A., Radimirsch, M., Vollmer, D.: Position-Aware Ad Hoc Wireless Networks for Inter-Vehicle Communications: the Fleetnet Project. Proceedings of the ACM Symposium on Mobile Ad Hoc Networking & Computing, MobiHoc 2001, Long Beach, California, USA (2001)
4. Lott, M., Halfmann, R., Schulz, E., Radimirsch, M.: Medium Access and Radio Resource Management for Ad hoc Networks based on UTRA TDD. ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001), Long Beach, CA, USA (2001)
5. Ebner, A., Rohling, H., Halfmann, R., Lott, M.: Synchronization in Ad Hoc Networks based on UTRA TDD. Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2002), Lisbon, Portugal (2002)
6. 3GPP TS 25.868 V5.0.1: Node B Synchronization for 1.28 Mcps TDD. 3GPP Technical Report (2002)
7. Akaiwa, Y., Andoh, H., Kohama, T.: Autonomous decentralized inter-base station synchronization for TDMA microcellular systems. IEEE Vehicular Technology Conference (1992)
8. Chuang, J.: Autonomous time synchronization among radio ports in wireless personal communication. IEEE Transactions on Vehicular Technology **43** (1994) 27–32
9. Hirukawa, A., Takanashi, H.: Inter-Base-Station TDMA Frame Synchronization Technique for Street Microcellular Systems. Proceedings 5th IEEE Personal, Indoor and Mobile Radio Communications (PIMRC 94) (1994)
10. Akazawa, H., Nakagawa, M.: Autonomous Decentralized Synchronization System for Inter-Vehicle Communication Network. Proc. 2nd World Congress on Intelligent Transport Systems (1995)
11. Sourour, E., Nakagawa, M.: Mutual Decentralized Synchronization for Intervehicle Communications. IEEE Trans. On Vehicular Technology **48** (1999) 2015–2027
12. Hübner, D., Hoff, S.: Simulative Evaluation of a Decentralized Packet Synchronization Algorithm for Short Range Mobile Radio Networks. Proceedings Globecom (1995)
13. 3GPP TS 25.221 V5.0.0: Physical channels and mapping of transport channels onto physical channels (TDD). 3GPP Technical Specification (2002)
14. 3GPP TR 25.928 V4.0.1: 1.28Mcps functionality for UTRA TDD Physical Layer. 3GPP Technical Report (2001)
15. Lombardi, M.A., Nelson, L.M., Novick, A.N., Zhang, V.S.: Time and Frequency Measurements Using the Global Positioning System (GPS). Proceedings of the Measurement Science Conference, Anaheim, CA, USA (2001)
16. Wheatley, C.: Self-synchronizing a CDMA cellular network. Proceedings Wireless Technologies China (1999)

17. Brown, R., Hwang, P.: Introduction to Random Signals and Applied Kalman Filtering. John Wiley and Sons, New York (1992)
18. Lombardi, M.A.: Frequency Measurement. The Measurement, Instrumentation, and Sensors Handbook, CRC Press (1999)
19. Allan, D.W.: Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control **UFFC-34** (1987) 647–654
20. Brandenberger, H., Hadorn, F., Halford, D., Shoaf, J.: High quality quartz crystal oscillators: frequency-domain and time-domain stability. Proceedings of the 25th Annual Symposium on Frequency Control (1971)
21. Chi, A.: The mechanics of translation of frequency stability measures between frequency and time-domain measurements. Proceedings of the 9th Annual Precise Time and Time Interval (PTTI) Applica-tions and Planning Meeting, Greenbelt, MD, USA (1977)
22. Ebner, A., Rohling, H., Lott, M., Halfmann, R.: Decentralized Slot Synchronization in Highly Dynamic Ad Hoc Networks. Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC'02), Honolulu, Hawaii (2002)
23. Nilsson, R., Sjöberg, F., Isaksson, M., Cioffi, J.M., Wilson, S.K.: Autonomous Synchronization of a DMT-VDSL System in Unbundled Networks. IEEE Journal on Selected Areas in Communications **20** (2002) 1055–1063
24. Ebner, A., Wischhof, L., Rohling, H.: Aspects of Decentralized Time Synchronization in Vehicular Ad hoc Networks. Proceedings of the 1st International Workshop on Intelligent Transportation (WIT 2004), Hamburg, Germany (2004) 67–72
25. Anderson, W.N., Morley, T.D.: Eigenvalues of the Laplacian of a graph. Linear and Multilinear Algebra **18** (1985) 141–145

2

Data Link Control

Matthias Lott[1], Rüdiger Halfmann[1], Egon Schulz[1],
Michael Meincke[2], Maria Dolorez Perez Guirao[2], and Klaus Jobmann[2]

[1] Siemens AG, Communications,
Sankt-Martin-Straße 76, 81541 München, Germany
{matthias.lott|ruediger.halfmann|egon.schulz}@siemens.com
[2] University of Hanover, IANT,
Appelstr. 9A, 30167 Hanover, Germany
{meincke|perez|jobmann}@ant.uni-hannover.de

Summary. This chapter investigates the adaptation of the cellular UMTS UTRA TDD datalink control to a VANET setting. The motivation for this is twofold. First, UMTS UTRA TDD provides very desirable characteristics, such as robustness to high relative speed between communication partners. Second, UMTS UTRA TDD hardware will become a mass-market product. Therefore this approach will reduce the cost to equip vehicles with the required radio hardware. To adapt the cellular UMTS standard to an air-interface for VANETs, changes of the medium access control sub-layer and radio resource management are required. An overview of the required modifications is given here. Performance results of the overall system considering throughput and delay are derived by means of analytical evaluations and event-driven simulations. Based on realistic mobility models, it is shown that the presented solution provides a robust communication platform even in extremely dynamic vehicular environments.

2.1 Introduction

Existing air interfaces that support self-organization such as the wireless local area networks (WLAN) IEEE 802.11 or HIPERLAN/2 [2] face significant challenges when employed in a VANET setting. In particular maximum communication distances of at least 1 km and support for operation at high relative velocities of up to 500 km/h [3] are hard to achieve with these systems. Within the standard IEEE 802.11p (WAVE)[3] the guard-interval of the orthogonal frequency division multiplexing (OFDM) scheme has been doubled compared to IEEE 802.11a to increase the multipath resistance and to combat higher Doppler-spreads [4]. However, high operation frequencies at 5 GHz still have a strong effect on the radio range. Moreover, speeds up to 250

[3] Currently, the air-interface for inter-vehicle communication is standardized in IEEE 802.11 in the task group TGp

km/h as allowed on German highways are still a big challenge with respect to inter-channel interference in an OFDM- based system. For HIPERLAN/2, fast network organization and reorganization is a major problem due to its centralized organization. In addition, it incorporates the same challenges as IEEE 802.11p.

As a consequence of those limitations an adaptation of the cellular UMTS UTRA TDD [5], [6] was investigated within the FleetNet project as a potential air interface for VANETs. UTRA TDD allows communication over long distances and supports high velocities. An additional benefit of UTRA TDD is the availability of an unlicensed frequency band at 2010 - 2020 MHz in Europe. Furthermore, UTRA TDD offers high flexibility with respect to asymmetric data flows and granularity for data transmission because of its code division multiple access (CDMA) component. It also supports QoS since it is possible to reserve transmit capacity owing to its frame and slot structure. In contrast to systems based on a WLAN standard, UTRA TDD was designed for a multipath propagation environment.

Since UTRA TDD was initially developed for operation in a cellular network structure, some modifications are required for an ad hoc operation. Besides the changes to Medium Access Control (MAC) and Radio Resource Management (RRM), also some modifications of the PHY layer are required to allow for an ad hoc operation. UTRA TDD comprises a Low Chip Rate (LCR, 1.28 Mchip/s) and a High Chip Rate (HCR, 3.84 Mchip/s) option. Since the LCR option is close to mass-market introduction, it was chosen as the key focus of this work.

Following a brief introduction to the air-interface of UTRA TDD this chapter presents the modifications required to adapt UTRA TDD to a VANET setting. The proposed solutions are then investigated via analysis and simulation.

2.2 UTRA TDD Radio Interface Protocol Architecture

The air-interface protocol stack of UTRA TDD is divided into three layers: the PHYsical layer (PHY), the Data Link Control Layer (DLC) and the Network Layer (NL), see Fig. 2.1.

The **PHYsical layer (PHY)** of the LCR mode defines a radio frame of 10 ms duration, which is divided in two sub-frames consisting of 7 time slots, respectively [8]. Following the first time slot, a special slot for synchronization is inserted, which is 275 μs long. Each of the regular time slots has the same length and contains the UTRA TDD LCR traffic burst, consisting of two data symbol fields of 352 chips, a midamble of 144 chips and a guard period of 16 chips. UTRA TDD allows the use of different channelization code lengths (spreading factor, SF: 1, 2, 4, 8, 16), which enables the parallel transmission of up to 16 codes in one time slot [9].

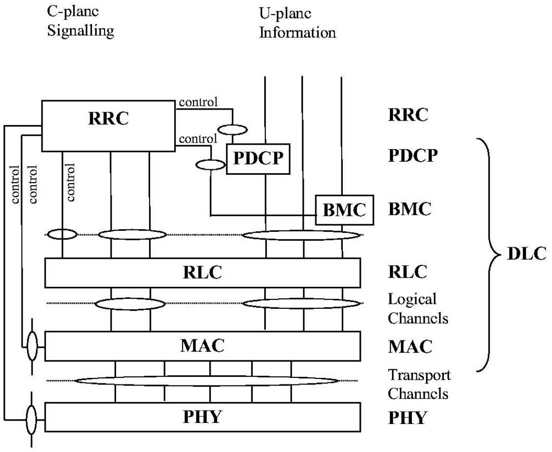


Fig. 2.1. Air-interface protocol architecture. The service access points are marked by circles.

The **Medium Access Control (MAC)** layer maps the logical channels of the RLC onto the transport channels supported by the physical layer. The priority handling between different data flows, which are mapped on the same physical resources, is also part of the MAC layer.

The **Radio Link Control (RLC)** layer provides transparent, unacknowledged or acknowledged mode data transfer to the upper layers. The main functions of the RLC are ciphering of data in acknowledged or unacknowledged mode and the support of an automatic repeat request scheme (ARQ) in acknowledged mode.

The **Broadcast/Multicast Control (BMC)** layer exists in the U-plane only. In the UTRAN, there is only one BMC entity per cell providing the services for cell broadcast. The Packet Data Convergence Protocol (PDCP) is present in the U-plane only. It provides header compression functions for network protocols, i.e. the Internet protocols IPv4 and IPv6.

The **Radio Resource Control (RRC)** layer as part of the NL handles the control plane signals between the UTRAN and the UEs. It is also responsible for controlling the available radio resources. This includes assignment, reconfiguration and release of radio resources as well as continuous control of the requested Quality of Service (QoS).

2.3 The Data Link Control in UTRA TDD Ad Hoc

2.3.1 State of the Art

Within the European research framework DRIVE, the use of communication technologies were investigated to improve the information distribution into vehicles [10]. The main focus was the integration of vehicles in cellular mobile radio systems, like UMTS, and broadcast systems, like DVB-T and the cooperation of cellular with broadcast networks, but not the inter-vehicle communication. Recent work on Roadside-to-Vehicle Communication (RVC) deals with the definition of vehicle-related services by using UMTS as communication infrastructure [1]. The distribution of locally restricted information, e.g. emergency messages upon accidents to inform subsequent vehicles, shall be realized by defining virtual networks of vehicles. The main drawback of the aforementioned investigations, however, is that due to the centralized network topology the messages have to travel through the air at least twice, from the vehicle to the base station and from the very same base station back to the other vehicles. Additionally, a central unit has to classify and process the messages before further distributing them to virtual car networks. This consumes unnecessary radio resources and introduces delays which may be decisive, e.g. for vehicles approaching an accident at high speed. In general, the centralized architecture is not efficient with regard to supporting applications that distribute data only between a group of mobiles which are spatially close to each other. Within 3GPP, an ad hoc mode called Opportunity Driven Multiple Access (ODMA) was proposed for UTRA TDD [11] as extension to the cellular mode to support direct- communication between mobiles. Different spreading codes and power control are used to cope with changing topologies and hidden-stations. However, due to its requirement of a central authority, ODMA is not suited for VANET. Moreover, distributed radio networks using CDMA have not gained major importance in the past and, hence, only few literature is available. The solution of the near-far problem resulting in an increased hidden terminal situation was recognized as the key challenge in such networks. Some elementary results in [12] state that the code assignment with only three available spreading codes is NP-complete. Therefore, only suboptimal algorithms are proposed. Another interesting MAC protocol proposal for a direct- sequence spread-spectrum system for VANETs was published in [13]. The reservation is performed once when a vehicle enters a network by probing all slots. If there is no reaction to the probe signal, the slot is occupied by the terminal. If this terminal senses a probe signal in its time slot, it sends a negative acknowledgement. Additionally, it is proposed to choose time slots and codes according to the so-called head spacing, which reduces the negative effect of the hidden-terminal problem. The problem with this proposal is that it is mostly based on polling, which we feel is inefficient for our purposes. A MAC protocol for a vehicle-based ad hoc network called Decentral Channel Access Protocol (DCAP) was investigated in the PROMETHEUS framework in the early nineties [14], which proposes a similar scheme as the one proposed here. It is based on a frame structure and reserves capacity by means of reservation ALOHA. To cope with hidden stations it is proposed to distribute bit maps where each station announces its occupancy status of all time slots via radio broadcast. However, it describes only a framework for a class of MAC protocols without going into deeper detail on, e.g., the used radio interface and the frame duration. Moreover, it is restricted to a single source- destination constellation per time slot, as opposed to the proposal in this paper. An extension to DCAP is described in [15], where the protocol overhead of the bitmap transmission is reduced and a handover procedure is proposed where an interfered station initiates a handover request, which indicates the affected slot. This indication is answered by a handover reply including the old slot number. Closely related to the approach proposed here, within CarTALK a variant of the MAC protocol based on UTRA TDD has been developed. This protocol is specialized for broadcast transmission [7] [16], [17], [18]. The protocol, named Reliable R-ALOHA (RR-ALOHA) protocol, has been defined in order to satisfy CarTALK 2000 system requirements and to support environments in which terminals can be grouped into clusters such that all terminals belonging to the same cluster experience broadcast radio communication among themselves. This cluster is defined as One-Hop (OH) cluster. At the physical layer it is also assumed that terminals belonging to different clusters can not communicate directly with each other. A terminal can belong to more than one OH-cluster leading to the case of non-disjoint clusters. The union of all OH-clusters having a common subset of terminals is denoted as a Two-Hop cluster. Based on the RR-ALOHA protocol, an ADHOC MAC protocol has been designed which operates on a time slotted channel and implements a Dynamic TDMA mechanism that is able to provide prompt access and variable bandwidth reliable channels as needed for QoS delivery. To fulfill this task, it is proposed to transmit some additional information, the Frame Information (FI), to let any terminal know the status (available or reserved) of each slot. So the basic idea is that each terminal periodically transmits the perceived status of slots in the preceding period (frame). The basic CarTALK system parameters, e.g. on the physical layer, the frame structure, etc., and the behavior is comparable and aligned to the approach developed within FleetNet, which is explained in the next sections.

2.3.2 Medium Access Control

In the UTRA TDD ad hoc mode developed within the FleetNet project, transmit capacity at one frequency is provided by a combination of one or more time slots and one or more codes. Such a combination of frequency, slot, and code provides transmission capacity for one logical channel, which can be used as unicast, anycast, multicast, or broadcast connection. One challenge inherited with the usage of CDMA is the power-impairment problem [6]. The power-impairment problem is characterized by the phenomena that for specific transmitter-receiver constellations the near- far effect cannot be resolved by means of power control. When assigning different spreading codes to different connections in a wireless system, the codes that are received at the same time need a minimum power difference to be detected. If joint detection is performed, as is the case with UTRA TDD, the reception power of the unwanted code should not exceed the wanted code by approx. 10 dB. In case a simple Rake receiver is used, this difference is even much smaller. Under several propagation conditions, the simultaneous transmission on different links cannot be realized by using different codes because of power impairments, as explained in the next Fig. 2.2 [6].

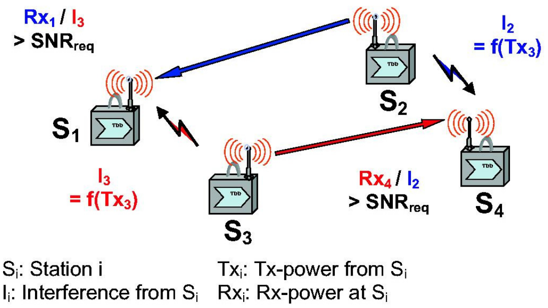


Fig. 2.2. Power-impairment problem for two connections

In the example, station 1 (S1) receives interference, I_3, from station 3 (S3) and thus requires higher transmit (Tx) power from station 2 (S2). As S2 increases its Tx power, S4 receives higher interference power from S2 (I_2). This in turn induces S3 to transmit with a higher Tx power to combat the interference at S4, but at the same time increases the interference at S1. Even optimal power control cannot resolve this communication relation. This conflict is denoted as power-impairment problem. The only solution for the above described case is to select a new time slot for one of the connections. To preclude the power-impairment problem that is associated to a CDMA component in an ad hoc network, it is proposed that only one station is allowed to transmit in one slot at the same time. With the proposed concept, several stations (up to the number of parallel codes that are supported, in case of UTRA TDD this number is 16) can be simultaneously reached by one station. The approach is equivalent to the downlink direction in a cellular CDMA system and, therefore, still exploits the advantage of fine granularity of capacity offered by the CDMA component over a pure TDMA system and the diversity gain inherited by CDMA. Although the proposed concept of code assignment takes into account the code domain, it requires no closed loop transmit power control

(TPC) for detection of codes that are transmitted in the same slot but with different spreading codes. Furthermore, a simple Rake receiver is sufficient for equalization and code separation, which reduces the hardware complexity at the receiver branch considerably. Another advantage of a TDMA system over e.g. a pure FDD or a CSMA system is the possibility to reserve an exclusive part of the frame for high-priority services. Reservation of transmit capacity is a basic requirement for quality of service (QoS) guarantees.

Support of Different Service Classes

In VANETs diverse applications with different needs for QoS exist. The different reservation methods of the proposed MAC protocol cope with the different classes:

- Emergency notifications are of enormous importance. They do not occur often but when they occur, sufficient capacity must be available. Therefore, at least one time slot in every frame is constantly reserved for this service. Access to this high priority channel is performed by a simple random access scheme like ALOHA, which will not be considered in more detail in this book. It is assumed that this channel is utilized very seldom and, hence, no special mechanisms for congestion avoidance is needed. The exclusive reservation of capacity for this kind of service is sufficient for the QoS guarantee.
- All other services may have delay or throughput constraints, but they are not as important as emergency notifications. Therefore, they share the remaining part of the MAC frame. The underlying services are called on-demand services in the following.

The resulting frame structure is shown in Fig. 2.3. A certain part of the capacity in terms of slots in a frame is constantly reserved for high priority services and the remaining part, called on-demand dynamic reservation phase, can be dynamically assigned and temporarily reserved by different stations for several services with lower priority.

The boundary between the two parts is flexible but defined by the slot structure. The minimum number of high priority slots, N_{high}, is a system parameter and assumed to be $N_{high} = 1$ in the following. Reservation ALOHA (R-ALOHA) is the basis for the reservation of capacity for on-demand services. It is well known that it has better performance than slotted ALOHA. Still, R-ALOHA has the potential risk of instabilities and may result in high collision rates for many participating stations and frequent reservation attempts, particularly in the case of short packets. Hence, a different reservation scheme shall be used where a certain share of the available capacity is permanently reserved, resulting in so-called ”circuit- switched” broadcast channel (CSBC). If the number of stations is not too high, each station reserves a CSBC. If the number of stations grows, only a certain ratio of them reserves a CSBC. All others use pure R-ALOHA.

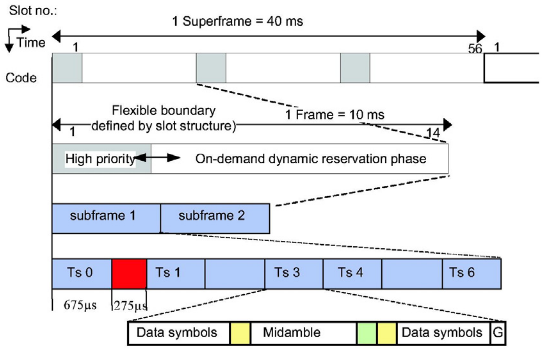


Fig. 2.3. Organization of the frame

Reservation of Transmit Capacity

For initial reservation of CSBCs, R-ALOHA is used (cf. step 1 in Fig. 2.4). To reduce the amount of permanently reserved capacity, a super frame is introduced, which comprises a number of basic frames. In the following it will be assumed that a superframe consists of 4 basic frames, resulting in a superframe duration of 40 ms (cf. Fig. 2.3). If a station has data to be transmitted, it may use its CSBC. If this capacity is not sufficient, it uses the CSBC to transmit a reservation for additional capacity (inband-signaling, cf. step 2 in Fig. 2.4). Reserved slots are sensed and will be respected by the neighboring stations. Packets transmitted in the same slot in subsequent frames can be described as packet train. The reservation for the used slot is maintained by piggyback signaling (cf. step 3 in Fig. 2.4) until the train ends and the slot is released. This results in a system behavior of R-ALOHA without collisions.

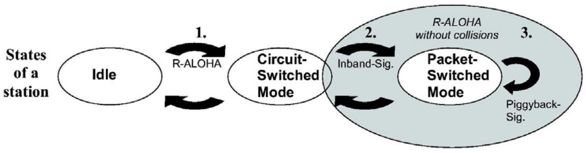


Fig. 2.4. States of the MAC protocol

Connection Management

After a packet has arrived from higher layers, the station waits until its next CSBC slot is available to reserve transmit capacity in this or in the next frame.

To avoid collisions with reserved slots of other stations, the usage status of all slots is measured before reservation. This decoding phase has the length of one frame duration (10 ms, 14 slots resp.) and helps to predict the usage status of the 14 slots in the following frame duration. In Fig. 2.5, the resulting frame structure and the usage of reserved slots is shown.

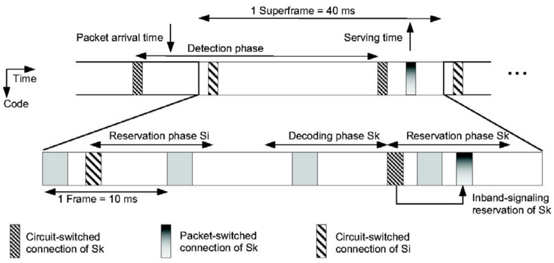


Fig. 2.5. Example for capacity reservation by means of inband-signaling

The reservation procedure of additional capacity is based on the knowledge which channels (time slots) are available, i.e. which are free of interference, and which are used (reserved) by other stations. This knowledge will usually be gained by a) measuring the radio channel, and b) receiving reservation packets from neighboring stations. Measuring the signal strength (RSSI, Received Signal Strength Indicator) of each time slot, a station can detect the status of each slot. If the RSSI is below a predefined threshold Th_{detect}, the channel is expected to be unused; if it is above Th_{detect} but below a second threshold Th_{decode}, it is used. If the signal strength is above Th_{decode}, the transmitted data can be decoded. In the latter case, a station can obtain the IDs of the communicating stations, detect reservation messages and release flags, i.e. it can forecast the usage status in the following frames. This forecast is needed in the case when a station wants to reserve additional resources without disturbing foreign reservation requests. However, even without decoding messages, the future reservation of slots can be anticipated because of the frame and slot structure and the fact that used slots are automatically reserved in the next frame (principle of R-ALOHA). This is a fundamental advantage of reservation-based MAC schemes using a frame/slot structure compared to asynchronous schemes that use random-access, e.g. CSMA. However, due to mobility and partially meshed networks there might result reservation conflicts, i.e., two stations might reserve the same slot for transmissions. One solution to resolve reservation conflicts will be to explicitly acknowledge reservations, i.e. if a reservation is received and packets can be correctly decoded by the receiver, it sends an acknowledgement (ACK) back to the reserving node. If the receiver detects a collision it will send a negative acknowledgement (NACK). While the CSBC will be used to send these (N)ACK packets, a maximum of 4 collisions per reserved resource (slot) may occur, depending on the position of the CSBC (cf. Fig. 2.6). If the CSBC of node 2 would be near the end of frame 2 (after the reserved slot), the reservation could be cancelled and only one packet would cause a collision.

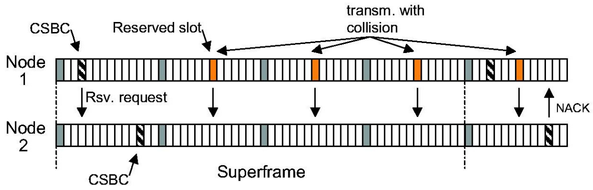


Fig. 2.6. Reservation acknowledgements

Nevertheless, collisions will occur due to reservation conflicts owing to hidden stations. To reduce this resource of potential inefficiency the impact of hidden stations on the reservation mechanism should be combated.

Mechanisms to Combat Hidden Stations

Hidden stations exist in partially meshed networks where some stations cannot reach all stations within one hop but need several hops to communicate. As consequence stations outside the range of a transmitter but inside the interference range of the target destination are called hidden stations (HS) [19]. When an HS starts transmission during an ongoing data transfer a collision will occur at the receiver. Due to node mobility and the presence of hidden stations, the knowledge of each station on the usage status of slots is not always complete. Hence, nodes may have reserved identical resources and packet collisions can occur. Collisions have to be resolved by reservation of new resources and lost packets have to be retransmitted, which leads to a degradation of system performance, i.e. larger packet delay and lower spectral efficiency. This hidden-station problem is known for a long time and solutions to mitigate this problem has been specified, e.g., in the standard IEEE 802.11. This standard specifies a reservation scheme comprising a Request-To-Send (RTS) signal transmitted by the sender, and in response a Clear-To-Send (CTS) signal transmitted by the intended receiver before the data transfer starts. By this the sender and receiver reserves transmission capacity and HS are informed by the CTS signal about the following data transfer. Another approach has been proposed in the Wireless- Channel Oriented Ad-hoc Multihop Broadband (W-CHAMB) network [20]. To avoid HS it is proposed to incorporate Time-Division-Duplex (TDD). Both stations, transmitter and receiver, share one time slot in a frame. The alternating usage of the slot by both stations informs all potential HS in the detection range of transmitter and receiver. A further improvement for the W-CHAMB foresees the assignment of energy signals at the end of each frame [2]. Each energy signal indicates the status of the corresponding slot in the frame, i.e., if there is transmitted an energy signal the respective slot has been used for reception, otherwise this slot is free. Different to the aforementioned approaches it is proposed for UTRA TDD ad hoc to explicitly broadcast the information on used resources (Channel State Indicator, CSI) to all neighbors via respective data packets. To efficiently spread this information, the respective packets are piggybacked to packets transmitted on the CSBC. This CSI can contain full information on the state of all channels (e.g. channel free / channel busy / interference detected). The CSI corresponds to the Frame Information (FI) proposed for the RR-ALOHA protocol developed in CarTALK [17], [18]. To reduce the overhead that is needed to inform the neighbors about the state of all 56 slots inside one superframe investigations have been carried out [21] and have shown that it is sufficient if each node spreads information on slots used for receiving to all of its neighbors. First, this will reduce the size of the table and will avoid transmission of redundant data. Stations will receive CSIs from all surrounding stations and can combine them to a more complete knowledge. Second, packet collisions happen at the receiver, so information on successful receptions are much more important than on potential unsuccessful transmissions.

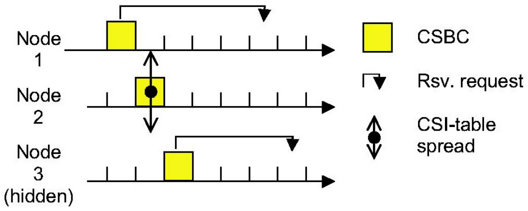


Fig. 2.7. Spreading of CSI to prevent from collisions

In the example depicted in Fig. 2.7, station 2 spreads its CSI containing information which slots it intends to use for reception in the future: here, the CSI includes the slot reserved by node 1. This would prevent node 3 from reserving the same slot. Additionally, because the CSI is received by node 1, this also acts as a reservation acknowledgement. This mechanism is slightly comparable to the RTS/CTS handshake mechanism of the IEEE 802.11 MAC: the resource reservation message inside the CSBC acts like a RTS and the ACK via the CSI acts like a CTS message. But by using the CSI, more than one reservation can be acknowledged by just one message, reducing overhead in comparison to a dedicated ACK sent by each intended receiver to every corresponding transmitter.

Efficient Broadcast

Broadcast is an important service for sending urgent messages, in which some nodes are chosen to relay the broadcast packets to reach all nodes. Plain flooding approach on the network layer has the drawback of generating excessive retransmissions, resulting in collisions and a highly inefficient use of bandwidth. In fact, for any packet that has to be broadcast to the entire network, a limited set of nodes should be chosen to relay the packet. In [22] a mechanism is proposed how stations decide whether they should re-broadcast a packet or not.
Let denote by C_i the set of neighbors of a station i ($C_{i=4} = \{S2, S5, S11\}$, and by C_j^i for any $j \in C_i$, the sets of neighbors‘ neighbors ($C_2^i = \{S3, S4, S10\}$, $C_5^i = \{S4, S6, S11\}$, $C_{11}^i = \{S4, S5, S12\}$), see Fig. 2.8. Since S2 is not in the decoding range of any neighbor of Si, it is excluded from the possible sets C_j^i (it is not a neighbors´ neighbor of Si) though he is part of the neighbors of Si, i.e. $S2 \in C_i$.

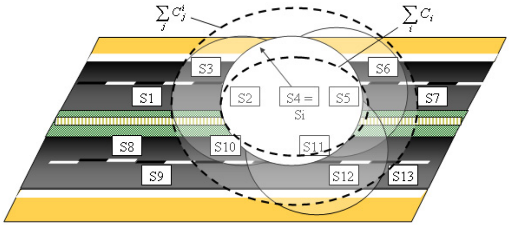


Fig. 2.8. Example scenario for broadcast

Given that station i (Si) receives a broadcast packet from station z, e.g. S5, in slot k, it is defined the set of neighbors of Si that have not received the packet in slot k by $U_i \subset C_j^i$ ($U_{i=4} = \{S3, S10, S12\}$). All these sets are identified by station i through the information carried by the frame information (FIs), i.e. busy/idle slot status, received in the N slots following slot k. In fact the set Ci contains all the terminals from which terminal i has received the FI, the

set C_j , for each $j \in C_i$, is identified by the entries in the FIs received by the terminal j, and the set U_i includes only the neighbors that have marked in their FI the slot k as IDLE and with the same identity of the transmitting terminal observed by i.
At slot $k+N$ station i recognizes whether or not it needs to relay the broadcast packets according to the following rule:
Station i does **not relay** the packet

a) if $U_i = 0$ or

b) for at least one $j \in (C_j^i - U_i)$ the following condition is satisfied:

$$U_i \subseteq C_j^i \quad AND \quad \big(|C_j^i| > |C_i| \; OR \; \big(|C_j^i| = |C_i| \; AND \; ID_j \geq ID_i\big)\big) \tag{2.1}$$

where ID_i denotes the address of station i. In the example in Fig. 2.8 $|C_2^i| = 3$, $|C_5^i| = 3$, $|C_{11}^i| = 3$ and $|C_4| = 3$ and $j \in (C_j^i - U_i)$ corresponds to $j \in \{4, 5, 6, 11\}$. Basically, station i does not relay the packet if set U_i is subset of C_j^i and if either C_j^i has higher cardinality than C_i or, having the same cardinality, the address of j is higher than that of i.
According to this rule only selected terminals will relay the broadcast packets thus significantly reducing the number of retransmissions with respect to flooding. Even if the optimality of this procedure is not guaranteed in the general case it is worthwhile noting that the minimum set of relaying terminals needed to cover the whole network is selected in most cases.

2.3.3 Radio Link Control

The Radio Link Control (RLC) layer takes over the task to make a secure data communication possible. In VANETs the RLC layer should work in transparent data transfer mode for emergency notifications but with acknowledged data transfer mode for all other services. Main functions of the RLC protocols are segmentation and reassembly, buffer management and, in the acknowledged data transfer mode, automatic repeat request (ARQ).

Automatic Repeat Request (ARQ)

Information transfer, especially over wireless channels, is subject to errors due to the unreliable communication channel. One mean in digital communication to combat errors on the channel is forward error correction (FEC). By introducing additional redundancy the corrupted symbols/bits can be corrected at the receiver. However, if too many bit errors occur, the correct information cannot be retrieved. An alternative is the re-transmission of faulty received information, the automatic repeat request. For the use of ARQ it is necessary, that each packet contains a sequence number (SN), so that both the transmitter as well as the receiver can identify a packet uniquely. The SN is limited in its size by way of its bit representation, which can lead to definiteness problems. Thus, rules are defined along with ARQ for the handling of transmission windows (sliding windows), which make sure the definiteness. Both, the transmitter as well as the receiver, take care of such transmission windows. A transmission window indicates, which packets along with which SN are allowed to be sent and received. The transmission windows are moved because of determined events. In the receiver this is generally the faultless reception of a packet, in the transmitter the faultless reception of an acknowledgment. Three different basic ARQ schemes used in fixed as well as in wireless systems are Stop- and-Wait (SW), Go-back-n (GBN) and Selective-Repeat (SR). More information about ARQ-protocols can be found in [2], [23], [24], [25], [26], [27]. A buffer management function is required for the ARQ protocol to enable retransmissions, in-sequence delivery of packets and to support several traffic priorities for respective VANET applications. For in- sequence delivery of packets, in case of using SR as ARQ scheme, reordering of packets in cooperation with buffering is required. The existence of several buffers allow to provide priority among different sorts of traffic, e.g. priority of ARQ messages or control packets used in reservation tasks over data packets, or priority of retransmissions over newly generated data packets.

ARQ in UTRA TDD

Within UTRA TDD ARQ is part of the Radio Link Control (RLC) layer (cf. Fig. 2.1). The acknowledged mode is only applicable to the dedicated control channel (DCCH) and dedicated traffic channel (DTCH) [28]. In the acknowledged mode of a TDD connection, the basic ARQ operation in release 99 of UTRAN is hybrid ARQ type I, which consists of a combination of forward error correction (FEC) and retransmissions. The receiving RLC requests retransmissions of lost PDUs by sending status PDUs to the transmitter to indicate which PDUs were received correctly and which PDUs require retransmissions. The applied selective reject (repeat) ARQ scheme is highly configurable. The relevant parameters are set by the Radio Resource Control (RRC) layer in the network during connection setup and are static during the connection if not reconfigured by the RRC. The receiver can be polled by the sender to transmit status PDUs, or transmits them if it detects missing PDUs. The status PDUs can be either transmitted in dedicated packets or can be piggybacked on ongoing transmissions. In UTRAN, a maximum transmit window size of $[0, 2^{12} - 1]$ can be used. The applied window size at the transmitter will be set by the receiver and can be changed during connection by sending special status PDUs. The selective reject ARQ scheme is realized by sending (negative) acknowledgments to the transmitting node. The acknowledgments can contain several types of data:

- Positive acknowledgment of PDUs with sequence numbers (SN) smaller than a specified SN (LSN, Last Sequence Number);

- Negative acknowledgment, using a list of sequence numbers, where consecutive PDUs not correctly received are reported (list super field);
- Negative acknowledgment, using a bitmap, where within a range of SNs the correctly and incorrectly received PDUs are reported (bitmap super field).

ARQ for UTRA TDD ad hoc Mode

In cellular networks, long connection times are available, large sliding windows and sequence numbers are used. In the case of highly dynamic ad hoc networks, the ARQ should not be controlled by a central instance, but it is equally distributed between communicating stations. Furthermore, it can be expected that the communication will not take part between one and several stations like in a cellular system, but between arbitrary stations in a peer-to-peer fashion. This will also impact the ARQ scheme in every station, which has to handle several links to different stations at the same time (and not only to one base station like in a cellular system). Existing ad hoc networks like IEEE 802.11 and Bluetooth implement a Stop-and-Wait mechanism, with positive acknowledgments (ACK) in IEEE 802.11 and negative unnumbered ACKs in the case of Bluetooth. Both systems dispose of a random access scheme to access the network, which demands fast ACKs of sent packets. Traffic characteristics in FleetNet are characterized by large packet trains between peer stations and collision free reservation requests for packet transmissions, which make fast ACKs not essential. Therefore, GBN and SR schemes can be used, and an improved error control performance can be achieved. Because of the special MAC scheme used in UTRA TDD ad hoc, the performance of the ARQ scheme differs from the one proposed in the standard. In UTRA TDD ad hoc it is proposed that every station keeps a special part of the available resources permanently reserved, the so-called Circuit Switched Broadcast Channel (CSBC). This constantly reserved capacity can advantageously be used as reverse channel for acknowledgments. But it can not be assured that a station will always and under any circumstances find available free space inside a CSBC channel to transmit its acknowledgments. FleetNet nodes will have to frequently exchange user data and control information for self-organization of the network. These information will include channel status tables where every station transmits its own knowledge about the channel status, reservation messages, and under particular traffic circumstances, periodical broadcasts of security relevant messages. All these information will be transmitted inside the CSBC. In a realistic approach, the existence of an available reverse channel to acknowledge sent packets is not assured. It is determined that after a mean waiting time of 20 ms (2 TDD frames) [6], the station‘s CSBC is available, assuming one CSBC very superframe of 40 ms. Thus, the station can use this transport capacity for sending the ARQ message. But the sum of the aforementioned messages leaves place only for short ARQ signaling reports inside the CSBC, as they arise with SW and GBN. Longer ARQ messages, as required in SR (e.g. using a UTRA TDD standard bitmap super-field with

3 octets, each bit representing a data packet, will require 44 bits of the total number of 210 bits available in the CSBC) demand an additional reverse channel. But this depends certainly on the number of packets which have to be acknowledged, i.e. on the protocol window size. Larger protocol window sizes mean, as well, larger delays from the high layer‘s point of view. In high traffic load situations, the average delay to get a reverse channel will increase, because the probability to get free slots decreases. Hence, there might be a trade-off in the required signaling effort and efficiency for SR and GBN, which might prioritize the selection of the one or the other protocol depending on the distribution of the channel errors and the error probability. Based on this arguments a performance comparison of both schemes GBN and SR has been carried out. The analytical and simulation results are described in sections 4.1.2 and 4.4.5.

2.4 Multiple Frequency Channel Exploitation

In UTRA TDD/LCR one frequency channel has a bandwidth of 1.6 MHz and provides a max. data rate of 1.28 Mbit/s in a frame of 10 ms comprising 14 time slots. For future applications, especially video and data applications, this data rate might not be sufficient, since this data rate has to be shared between all users using this frequency. And even for the target VANET applications that comprise delay-sensitive data, like emergency notifications, in parallel to Internet access, chat and transmission of pictures between vehicles, a higher data rate is envisaged. Furthermore, the available unlicensed frequency band between 2010 MHz and 2020 MHz allows to providing 6 frequency channels in parallel (cf. Fig. 2.9).

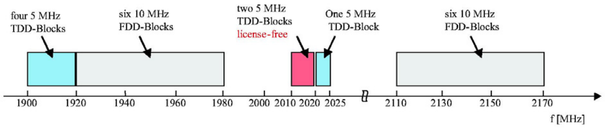


Fig. 2.9. German spectrum allocation for UMTS

To exploit the available frequency channels and to provide sufficient transmission resources for the targeted and future applications the Frequency Division Multiple Access (FDMA) component has to be made available for VANET. In this chapter appropriate algorithms and protocols for FDMA are provided. Based on the MAC protocol provided in the previous chapter, the management of available frequency resources in the system is the key issue of this chapter. The UTRA TDD standard specifies a

CDMA/TDMA/FDMA/TDD system. Hence, the management of the code, time and frequency domain has to be organized. The first version of UTRA TDD ad hoc has assumed to provide only one frequency carrier and a proposal for the CDMA/TDMA/TDD mode has been described. However, the potential of using different carrier frequencies should be deployed. The random access to acquire new resources is a key problem in wireless systems, which becomes more complicated in ad hoc networks where no central instance exists that can, e.g., provide much simpler mechanisms for collision resolution as compared to networks without a central instance. In combination with different frequencies this challenge becomes even harder since stations transmitting or receiving on one frequency might not be able to receive simultaneously on a different carrier frequency and might loose important signaling and reservation information. The questions how to support several carrier frequencies in an ad hoc network for vehicle-to-vehicle communication will be addressed and appropriate solutions are derived.

2.4.1 Challenges

With the distributed assignment of different frequency channels for VANET the following requirements arise, especially if only one transceiver is desired:

- Broadcast is the most important service to operate a wireless system. If stations are tuned to different frequencies a broadcast that is relevant for all stations should be sent on all frequencies. This might increase latency, reduce reliability, and increase the amount of transmitted data. This becomes even more critical for safety-related messages.
- There is a requirement to reach / access any arbitrary station for information exchange. If stations change their frequency the resources on which a station can be reached should be made available to the other stations. This requires considerable management efforts. Moreover, for dynamic allocation of resources different carrier frequencies and respective slots have to be measured and the status (reserved, free, collision, etc.) has to be monitored. At the same time blind slots make this task even harder. Blind slots occur when a station transmits / receives on one frequency, the other frequencies cannot be measured or used at the same time with affordable hardware complexity, e.g. with one transceiver only.
- To exploit the different frequencies time is needed for frequency switching.
- If more than one carrier frequency is used for transmission but only one frequency can be decoded at one time instance, receptions of transmissions as well as resource reservations without conflicts can not be assured. E.g., the setup of a new connection might fail if the target station is tuned to another frequency.

Though this section does not cover all challenges, even from these requirements it becomes obvious that a further degree of freedom makes the organization and a solution much more complicated.

2.4.2 Decentralized Frequency Division Multiple Access - dFDMA

The decentralized FDMA (dFDMA) concept foresees to logically subdivide time into two different phases [29]. During the first phase, the so-called *exchange phase* (EX-phase), a station listens and transmits on a predefined frequency, the coordination frequency, f_{coord}. On this frequency a station can announce reservation requests, send out beacons for the network organization, broadcast relevant or time-critical information to neighbors, exchange signaling information to run the protocol and manage the radio resources in a decentralized but controlled manner, etc. During the second phase, which will be referred to in the sequel as *arbitrary transmission phase* (AT-phase), the station is no more restricted to one carrier frequency. The station is rather allowed to arbitrarily exploit all available frequencies, f_i. During the AT-phase the station only has to take into account its communication relationships, and has to measure and test the available resource units that it is currently using and that it might use at a later time for transmission or reception. Since all frequencies in parallel to the coordination frequency cannot be used by stations that have only one transceiver and that are in the EX-phase, different frequency patterns are introduced, which define concurrent exchange and AT-phases. The phases of the frequency patterns are equidistant in time. An example for three frequency patterns is shown in Fig. 2.10.

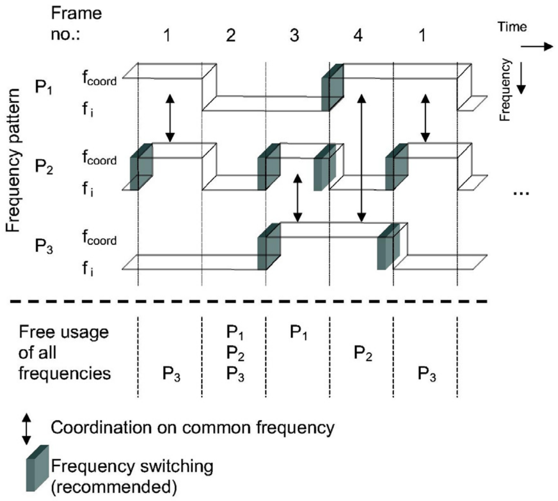


Fig. 2.10. Selection of frequencies in dFDMA, based on frequency patterns

Each frequency pattern will be repeated after four phases. For example, a station using frequency pattern one, P_1, will start with the EX-phase, i.e. operation on the coordination frequency, switches then for two consecutive phases to one or more arbitrary frequencies (AT-phase), and afterwards switches back to the coordination frequency, before it starts from the beginning. Advantageously, one phase corresponds to one frame if the dFDMA concept is applied to a framed system. The stations belonging to other patterns, P_2 and P_3, switch to the coordination frequency at different times, e.g., in case of P_2 in the first and third frame, and for P_3 in the third and fourth frame. Nevertheless, there are common times where stations belonging to two different patterns are jointly using the coordination frequency, e.g. frame 1 for the patterns P_1 and P_2, see Fig. 2.10. It is therefore guaranteed that two pattern, P_i and P_j, always have a common phase within four sequential phases, respectively frames. This guarantees a delay of four phases for the exchange of any information between stations within the decoding range (and as long as no channel errors occur).

Optimal Frequency Pattern

A station can join one of the three possible patterns. The number of three patterns is the minimum number that supports the permanent exploration of all frequencies with one transceiver only. Moreover, it is the maximum number of possible patterns with a period of four phases and no more than two EX-phases. There exist 24 possible combinations of EX- and AT-phases. Since at least two EX-phases are needed to communicate with stations of the two other patterns the number is reduced by 5 (only AT-phases and all possible combinations with one EX-phase only). Further 5 examples are excluded if no more than two EX-phases are allowed to support a max. number of AT-phases for each pattern. Only half of the remaining 6 patterns will be possible, since the complementary patterns of the 3 patterns shown in Fig. 2.10 will have disjunctive EX- phases, which makes an information exchange between stations of these patterns impossible. Of course more patterns can be introduced with more EX- and AT-phases, e.g. four patterns with a minimum of three EX-phases, but they will increase the latency of common EX-phases, respectively the repetition interval, and is therefore left out of consideration. Furthermore, the length of the EX-phase is the largest possible common duration for all patterns, which allows to exploit all available resources (all frequencies) at every time. An extension of the EX-phase leads to overlapping EX-phases for all three patterns that in turn precludes the exploration of other frequencies. A reduction of the EX-phase is possible with the drawback of reduced time to exchange broadcast / multicast messages. Hence, it can be followed that the example of the three frequency patterns depicted in Fig. 2.10 represents the optimal distribution of exchange and AT- phases. For dFDMA two different allocation schemes for the destinations exist:

1. **diff**: Source and destination belong to different frequency-patterns (except on the common coordination frequency they only can communicate during frame 2, where they can exploit all possible frequencies).

2. **com**: Source and destination belong to one common, i.e. the same frequency-pattern.

Impact of Frequency Switching Time

Considering the frequency pattern as depicted in Fig. 2.10, a station that has selected the pattern P_1 needs the last slot in frame 3 to switch to the coordination frequency in time, as far as we assume a frequency switching duration of one slot. The stations using P_2 and P_3 are also using the coordination frequency at that time. Hence, the other frequencies cannot be exploited. To avoid this potential waste of capacity an exception rule for the EX-phase is applied for stations of pattern P_2. They are allowed to switch to any other frequency 2 slots before the 3^{rd} frame ends. Hence, they start 2 slots earlier the AT-phase. With this modification that is applied to the other frequency pattern at the respective times, too, all slots at any frequency can be used if required, and all available resources will be exploited. At the same time the EX- phases are shortened by two slots. The resulting recommended frequency switching times are shown in Fig. 2.10.

2.4.3 Resource Reservation and Release for dFDMA

Before a station may reserve resources, it has to gain knowledge about the present status (used / reserved / free) of all desired resources. This is basically done by overhearing of reservation messages and data transmissions, i.e. scanning of available frequencies and slots. If a FleetNet station is equipped with a single transceiver, it can only work on one frequency at a single point of time. Introducing the dFDMA approach in VANETs, a station can chose out of all available frequencies in the AT-Phase and has to work on the predefined coordination frequency f_{coord} in the EX-Phase. Thus, a station is not able to gain full knowledge about the channel status on all frequencies by scanning only within one superframe. The resource reservation procedure works as follows, cf. Fig. 2.11:
Each active station has two CSBCs reserved during the EX-phases. Stations of pattern P_1 have one CSBC in sub- frame 1 and 4 each, stations of P_2 in sub-frame 1 and 3, and stations of P_3 in sub-frame 3 and 4. Exemplary selecting a station Si using pattern P_2 it will reserve one CSBC in sub-frame 1, e.g. slot 2, and slot 3 in sub- frame 3. Let us assume that Si (using pattern P_2) wants to reserve a slot on frequency f_2 to communicate with station Sk in frame 2 in slot 6. Si transmits a reservation request message inside its CSBC in frame 1. This message will contain the slot number, $sl6$, the frame number 2, and the frequency f_2. The receiver, Sk, will switch in the next frame (frame 2) for slot 6 on frequency f_2. The receiver Sk will transmit an ACK or negative ACK (NACK) in its next CSBC (in case of Sk in frame 3) to indicate to the transmitter if the reservation was successful and the packet(s) has(have) been

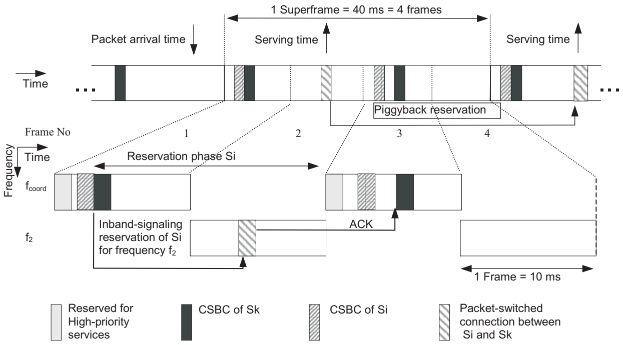


Fig. 2.11. Example for capacity reservation by means of CSBC using dFDMA

received without errors. The successfully reserved resource can now be used in subsequent superframes until it is released.

Since explicit piggyback reservations inside the data channels are quite hard to detect, because we cannot ensure that all other stations listen to all used frequencies, the piggyback reservations are also implicit, i.e. if a station reserves slot x on frequency y this means that this reservation is valid in every superframe (slot x, freq. y) until it is released again (traditional R-ALOHA). Besides, we provide a piggyback reserve / release flag in every data channel. This gives overhearing stations the information, whether the slot is used in the next superframe or not and allows them to access the channel before other stations, introducing some kind of prioritization and avoiding resource wasting. The release of a reserved resource is also done by the piggybacked release flag to indicate the end of a reservation. Even in a fully meshed network with no hidden stations reservation conflicts may occur. During the EX-Phase, each station has two CSBC slots in every superframe to transmit reservation messages for slots in a following frame on an arbitrary frequency. In every frame, stations belonging to the same frequency pattern and stations belonging to one other pattern (e.g. stations belonging to pattern P_2 and P_1 in frame no. 1, cf. Fig. 2.12) are able to overhear the reservations. However, stations belonging to the third frequency pattern (e.g. stations belonging to pattern P_3 in frame no. 1) are not aware of the reservation since they operate on a different frequency to exploit all channels at every time. These stations might propose the same slot in their reservation messages (e.g. in the previous superframe in frame no. 4), which has not been recognized by stations of P_1 and, consequently, results in a reservation conflict and might cause data packet collisions in a following frame (in the example in frame no. 2). Additionally, reservations can be irresolvable by the receiver. This happens, e.g., if an intended receiver has already a resource reserved on a different frequency than the frequency proposed by the transmitter in its reservation. This does not result in a packet collision, but in a packet loss anyway.

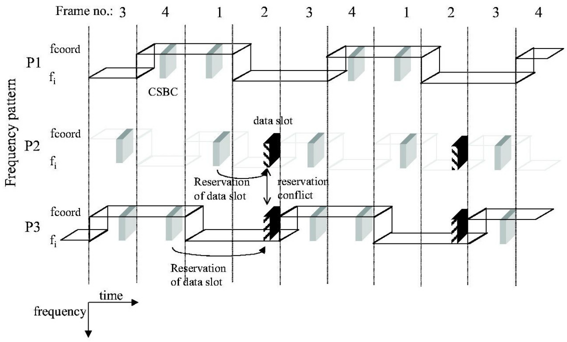


Fig. 2.12. Reservation conflicts in dFDMA

Reservation conflicts can be avoided, if we repeat reservations for slots in frame 2 in the second EX- Phase, e.g. station Sk has to repeat the reservation (first sent in frame 3) in frame 1 so that Sl can overhear it. This will increase the overhead that is needed for the reservation procedure. Another method to resolve reservation conflicts is called the CSI-table approach. If every station transmits its own knowledge about the channel status inside its CSBC in terms of channel status indicator (CSI), we can improve the system performance. The CSI contains information about all slots a station will receive a packet in the current superframe. The information comprises the slot number and respective frequency channel. The transmission of CSI tables can also help to solve the hidden station problem. If a station broadcasts its knowledge about used channels, stations in the vicinity will benefit from that. For additional information on the resource allocation problems refer to [21]. For performance results, considering esp. the resource allocation problems, refer to chapter 2.5.4.

2.5 Performance Analysis

If not explicitly stated, the following assumptions are valid for all performance investigations: A frame with duration of $T = 10\ ms$ comprising $N = 14$ slots is considered, whereby $N_{high} = 1$ slot is permanently reserved for high priority services. Simultaneous transmissions of more than one station result in collisions. No capture-effect is taken into account. All M stations have identical message arrival statistics that follow a stationary Poisson process with a message arrival rate of λ. Each message contains an average number of packets. One packet can be served within one slot. Each station has infinite buffering capacity. The station transmits in average packets before it gives up the reserved slot. In the following Table 2.1 the used variables and parameters are summarized.

Table 2.1. Simulation parameters

Symbol	**Value**	**Description**
N	14	Number of slots in frame
T	10ms	Frame duration
M		Number of stations
n_{SF}		Number of frames per super-frame
λ		Arrival rate of messages at station
$\overline{h}$		Average number of packets within a message
$\overline{\nu}$		Average duration of busy period

2.5.1 Analytical Description

Medium Access Control

The analytical model for performance estimation of UTRA TDD ad hoc mode is based on the model derived for R- ALOHA with constant throughput assumption [30]. This model has been selected because it decouples the basic analysis of the R-ALOHA protocol from the specific details of the contention protocol. In the analytical performance evaluation, it is assumed that the actual data transmission happens only in the N_d remaining slots not used for the CSBC. N_d depends on the number of active stations, M, each reserving one CSBC in every super-frame, and the slots in one frame, N.

$$N_d = N - M/n_{SF} \tag{2.2}$$

With the average delay, $\overline{d_A}$, for a successful transmission of a packet in a non-reserved slot the average message delay for the steady state can be determined for the proposed access scheme with the following formula [30].

$$d = \frac{\overline{x_0}}{1 - \lambda(\overline{x} - \overline{x_0})} + \frac{\lambda(\overline{x_0{}^2} - \overline{x^2})}{2[1 - \lambda(\overline{x} - \overline{x_0})]} + \frac{\lambda\overline{x^2}}{2[1 - \lambda\overline{x}]} \tag{2.3}$$

The mean service time of customers who initiate busy periods is

$$\overline{x_0} = \overline{d_A} + (\overline{h} - 1)T \tag{2.4}$$

the mean service time of customers who arrive to find the queue busy is

$$\overline{x} = \overline{h}T \tag{2.5}$$

and the respective second moments are

$$\overline{{x_0}^2} = \overline{{d_A}^2} + 2\overline{d_A}(\overline{h} - 1)N_d + (\overline{h^2} - 2\overline{h} + 1){N_d}^2, \tag{2.6}$$

and

$$\overline{x^2} = \overline{h^2}{N_d}^2. \tag{2.7}$$

This result has been derived under the assumption that each user queue can be considered as a generalized M|G|1 queue in which the first customer of each busy period receives exceptional service. Because of the different access method, the derivation of the access delay, $\overline{d_A}$ and the second moment, $\overline{{d_A}^2}$, differ from that of pure R-ALOHA, and are given by [6]

$$\overline{d_A} = \overline{T_{A,first}} + n_A T_{A,retry} + \overline{T_{A,trains}}, \tag{2.8}$$

with the average waiting time, $\overline{T_{A,first}}$, until the first reservation attempt that is given by

$$\overline{T_{A,first}} = 0.5 * n_{SF}T \tag{2.9}$$

and the waiting time, $T_{A,retry}$, between any subsequent reservation attempts that is given by

$$T_{A,retry} = n_{SF}T \tag{2.10}$$

For the estimation of $\overline{T_{A,trans}} = 1/2T$ we assume that the available slots are uniformly distributed over one frame resulting in an upper bound for the delay. The total number of reservation attempts to transmit one packet, n_A, can be derived under the assumption that a successful access attempt is only possible if at least one time slot is available in the frame following the slot of the CSBC. The probability that all slots are occupied is given by $P_{N_d} = U^{N_d}$, with U denoting the throughput of the system. Since the system is assumed to be in equilibrium, the channel throughput rate must be equal the channel input rate, and thus

$$U = M\lambda\overline{h}T/N_d \tag{2.11}$$

With the probability for a successful transmission $P_d = 1 - P_{N_d}$, the first and second moment of the access delay, $\overline{d_A}$ and $\overline{{d_A}^2}$, can now be derived:

$$\overline{d_A} = \frac{T}{2} + \overline{T_{A,first}} + \frac{1 - P_d}{P_d}T_{A,retry} \tag{2.12}$$

and

$$\begin{aligned}\overline{{d_A}^2} = &\left(\frac{1}{3} + \frac{n_{SF}}{2} + \frac{{n_{SF}}^2}{3}\right)T^2 + \frac{(1 - P_d)(2 - P_d)}{{P_d}^2}T^2_{A,retry} + \\ &(1 + n_{SF})\frac{1 - P_d}{P_d}TT_{A,retry}\end{aligned} \tag{2.13}$$

With these formulas, the average message delay for the proposed protocols can be determined. For a mathematically tractable analysis of the proposed protocol, it has been assumed that the CSBC is used for inband-signaling purposes only. In contrast to R-ALOHA, the inband-signaling on the CSBC is used to reserve resources without collisions. This is considered in Equation (8) by the average initial access delay, which incorporates the waiting time until a slot is available to be reserved in temporary situations of high traffic load. By means of this access mechanism, there is no need for an adaptive control algorithm for contention (a back-off parameter or load dependent access probabilities) like in R-ALOHA. At the same time, this scheme allows to exploit the free slots in an efficient way. For fairness reasons, it is foreseen that a station is forced to release a slot after a pre-defined duration, which is assumed to be ten super-frames, if all slots in a frame are reserved. This results in a trade-off between reservation overhead and delay. Alternatively, a station with low-priority traffic can be requested to release a slot by means of a respective message sent in the CSBC as described in [19].

Validation of Analytical Description of the MAC Protocol
For the purpose of validation of the analytical formula a simplified event-driven simulator based on the C++ class library CNCL (Communication Networks Class Library, [23]) has been developed. It incorporates the MAC protocols presented in the previous chapters and takes into account the different protocol states a station passes through as well as the reservation status of the slots in a frame.

In Fig. 2.13, the curve on the left indicates the mean delay for the case that the CSBC is used for signaling only.

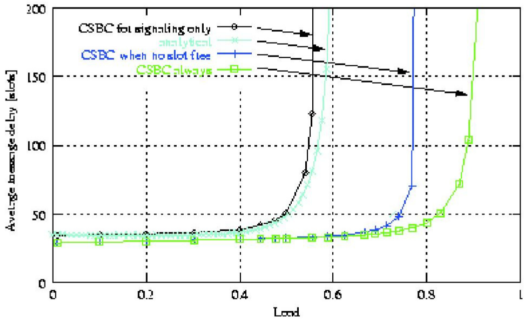


Fig. 2.13. Comparison of analytical and simulated results and results for different utilization of the CSBC (N=20)

The analytical derived curve and the one derived by means of simulations fit very well. Only for high loads in the point of saturation the curves show slight differences, which might result from the finite simulation times. If the network is highly loaded and no slot is free that can be reserved by means of inband-signaling, it is beneficial if the capacity of the CSBC is used to transmit the message instead. This is indicated by the middle curve in Fig. 2.13. With this modification, the achievable throughput for 20 stations and a mean delay of 50 slots can be increased from approximately 50 % to 75 %. The throughput for a delay of 50 slots, which corresponds to less than 40 ms, can be further increased if the slots reserved for the CSBC are used whenever packets have to be transmitted. The achievable throughput for acceptable delays becomes approximately 85 % and indicates an efficient operation of the proposed protocol. The difference to 100 % can be explained by the overhead that is still needed for the inband-signaling on the CSBC (no user packet can be transmitted on the CSBC when an inband-signaling packet is transmitted to reserve a slot). In addition, a small amount of unused slots reserved for the CSBC when no message is to be transferred by the respective station will exist.

Automatic Repeat Request

This section focus on the suitability of different ARQ methods for the ad hoc mode of UTRA TDD for VANETs. We compare analytically two possible ARQ methods for VANETs: Go-Back-N (GBN) and Selective Repeat (SR). For SR-ARQ the partial bitmap signaling mechanism of the UTRA TDD standard is used [28]. The impacts of the aforementioned ARQ schemes on the protocol overhead are presented. The overhead associated with an ARQ scheme can be decomposed into re-transmissions overhead and signaling overhead. We define the overhead efficiency of an ARQ protocol as the quotient between the error-free received bits and the total number of sent bits, i.e. user data, retransmissions and ARQ messages. The number of bits required to encode an ARQ message depends on (i) the RLC layer protocol packet format specifications, which in FleetNet are based on the UTRA- TDD standard, and determine the structure of the packets that will contain the ARQ information. It also depends on (ii) the signaling mechanism used, i.e the way the receiver announces to the sender which packets have been received erroneous, e.g. using a list super field or a bitmap super field. Finally, it depends on (iii) the concrete ARQ protocol configuration, i.e. on the window size. For our investigations we have assumed that not more than one error occurs in the same frame, i.e. the errors are distributed over the frames and no burst errors are taken into account. Moreover, we have distinguished two different scenarios shown in Fig. 2.14: a) where one user reserves only one additional slot every frame for data transmission, and b) where one user occupies the whole frame resulting in the maximum possible throughput.

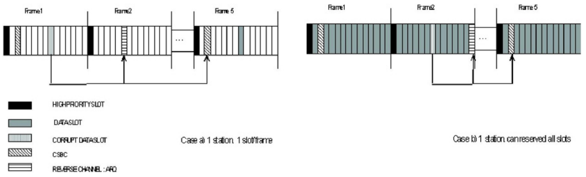


Fig. 2.14. Single user scenarios a) and b)

One slot per Frame

We have derived the following formulas to calculate the overhead efficiency of SR (S_{SR}) and GBN (S_{GBN}) for one active user for case a):

$$S_{SR} = \frac{L - 1 - \left(\frac{L}{WinSize} + max(1, \frac{WinSize}{BitmapSize})\right)}{L}, \quad PER \neq 0\% \tag{2.14}$$

$$S_{SR} = 1 - \left(\frac{1}{WinSize}\right), \quad PER = 0\% \tag{2.15}$$

$$S_{GBN} = \frac{L - N_{following} - \left(\frac{L}{WinSize} + 1\right)}{L}, \quad PER \neq 0\% \tag{2.16}$$

$$S_{GBN} = 1 - \frac{1}{WinSize}, \quad PER = 0\% \tag{2.17}$$

with L being the average number of frames until an error occurs, and $N_{following}$ denoting all sent packets following the corrupted packet until the ARQ message is received and which have to be retransmitted in case of GBN. The window size can take values $WinSize \geq 2$, whereas for $WinSize = 2$ the GBN protocol becomes a Stop- and-Wait ARQ protocol. As mentioned earlier, slots for ARQ signaling will be reserved only when necessary. This implies that the number of slots reserved for ARQ depends directly on the window sizes. For a fixed number of packets sent, the number of ARQ messages will increase with shorter window sizes since the reception window will be filled more often. With the GBN protocol it will be necessary to sent $\frac{L}{WindowSize} + 1$ ARQ messages, which corresponds to the same number of slots (slot granularity). We assume that one ARQ message (equal to one slot) can only contain one Bitmap Super field that can acknowledge only 128 packets. On that account, if SR works with window sizes larger than 128, more than one slot would be necessary to acknowledge a full window. Concretely, $\lceil \frac{WindowSize}{BitmapSize} \rceil$ slots would be necessary, being $BitmapSize$ equal to 128 and assuming that window size is always a power of 2.

The performance has been calculated considering the overhead efficiency, cf. Fig. 2.15. A fixed delay to get a reverse channel is assumed, concretely

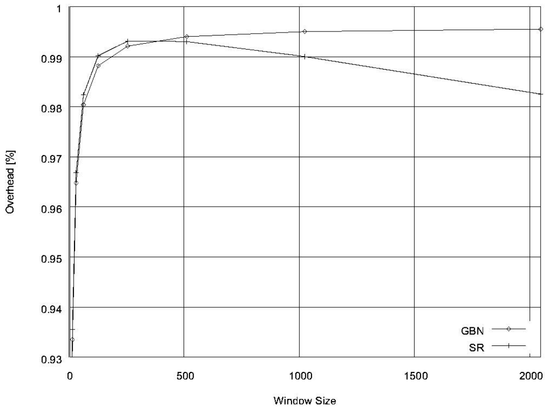


Fig. 2.15. Efficiency comparison, packet error rate $10e^{-3}$. Single user case a) with one slot per user per frame

it is assumed that the ACK/NACK fits inside a CSBC which in mean is available after 20 ms. The results show that for window sizes equal 256 the efficiency of Selective-Repeat is slightly better than that of Go-back-N. This small advantage is further reduced as the PER decreases. For window sizes > 256 more than one slot is needed to acknowledge a full window and the performance of Selective-Repeat is worse than of Go-back-N.

Full Frame The formulas for the protocol overhead efficiency of both ARQ schemes in the case a) can be used for the case b) with little modifications resulting in:

$$S_{SR} = \frac{N_{data} * L - 1 - \left(\frac{L}{WinSize} + max(1, \frac{WinSize}{BitmapSize})\right)}{N_{data} * L}, \quad PER \neq 0\% \tag{2.18}$$

$$S_{SR} = 1 - \left(\frac{1}{N_{data} * WinSize}\right), \quad PER = 0\% \tag{2.19}$$

$$S_{GBN} = \frac{N_{data} * L - N_{following} - \left(\frac{L}{WinSize} + 1\right)}{N_{data} * L}, \quad PER \neq 0\% \tag{2.20}$$

$$S_{GBN} = 1 - \frac{1}{N_{data} * WinSize}, \quad PER = 0\% \tag{2.21}$$

with N_{data} referring to the number of slots per frame that a user can utilize for data transmission, i.e. 14 slots in case b).

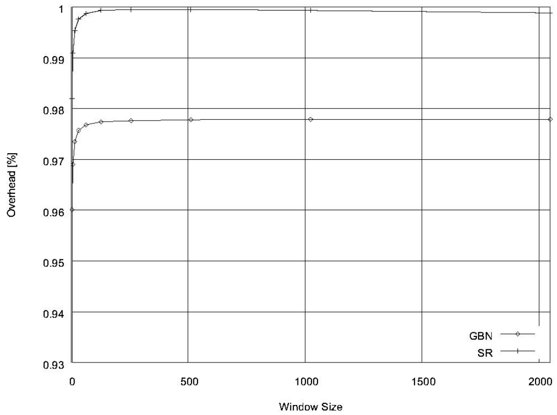


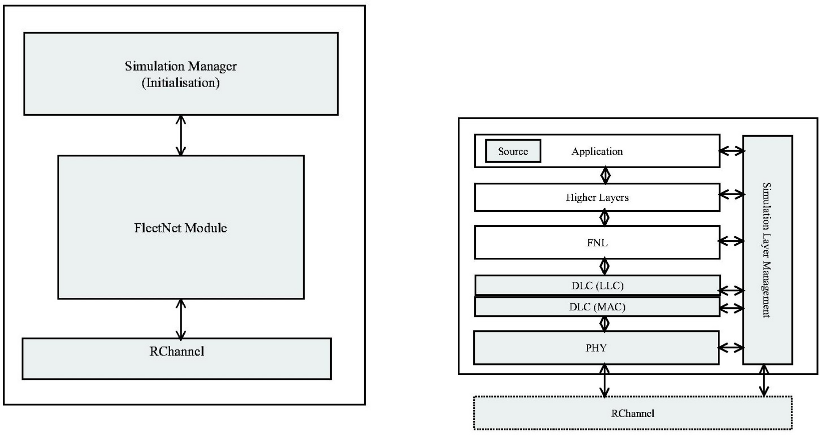
Fig. 2.16. Efficiency Comparison, packet error rate $10e^{-3}$. Single user case b) with several slots per user per frame.

The difference between the efficiency of both ARQ schemes is now more perceptible (cf. Fig. 2.16). This effect is because a larger number of packets must be retransmitted in the GBN case, but the difference is still small, i.e. less than 2.5%. With decreasing PER both protocols tend to show equivalent performance, e.g. for $PER = 10^{-5}$ both protocols have an overhead efficiency of more than 99,9%. Comparing both protocols, GBN and SR, it has to be mentioned that GBN has less overhead in the case of low PER and, consequently, the ARQ messages can be piggybacked with other data, e.g., in the CSBC. Different to GBN, the SR-ARQ protocol needs almost a whole slot, e.g. the CSBC, and hence, provides not the flexibility that GBN offers. Therefore, the GBN-ARQ protocol seems to be better suited for VANETs than SR. It offers almost the same efficiency as SR plus an additional flexibility and protocol simplicity, which also reduces the hardware requirements, e.g. packet queues.

2.5.2 Event-Driven Simulations

Simulation Environment

This section describes the structure and the implementation of a FleetNet Protocol Simulation Environment. The simulation system is built in SDL (Specification and Description Language) and realizes an event-driven simulation. The simulator consists of a number of generic parts, such as the simulation manager and the Radio Channel, and provides a framework for the implementation of the actual protocol and system. The structure of the FleetNet simulator and the structure of the protocol package (FleetNet Module) is given in Fig. 17(a), resp. Fig. 17(b).



(a) Structure of the Simulator

(b) Structure of the FleetNet module

Fig. 2.17. Structure of the Simulation Environment

The simulator provides a horizontal structure with a number of blocks representing the layers of the FleetNet reference model, a simulation manager, a radio channel and a device manager (FN_Manager). The simulation manager and the radio channel are mostly generic and are defined in the system. All other gray blocks were filled by the implementers in the course of time. The simulation manager provides a generic framework for the implementation of a radio network simulator for a certain system. It contains some generic functions to control the simulation, basically the initialization process, which provides the generic initialization routines. It reads some generic initialization parameters, i.e. it allows to determine the generation and removal of devices where the time to generate or to remove a device is given in the instruction. The Radio Channel covers the special characteristics of the channel we deal with within VANETs. Implemented basic functions will be a) interference measurement for Dynamic Channel Allocation (DCA) and b) calculation of path loss and interference. Different path loss models can be configured (e.g. models described in [31] or [32]). The FN_Manager block is responsible for transferring the signals from the simulation manager to the different layers and vice-versa. Therefore, most signals are converted to suitable signals for the specific blocks. The Device Manager is located within the FN_Manager and is responsible for the handling of any devices, e.g. FleetNet Nodes, Gateways etc. Especially, it contains functions for the movement of devices. The movement is controlled by movement models which will be generated out of

simulated traffic scenarios.
Only the gray layers of the protocol stack (cf. Fig. 17(b)) are implemented in the simulation system and will be described in the following.

- The Application Layer hosts the traffic sources. The traffic can be generated following a Poisson process with configurable packet size and arrival rate λ. Also constant bit-rate (CBR) sources are possible.
- The Data Link Control (DLC) layer is sub-divided into Logical Link Control (LLC) and Medium Access Control (MAC). The LLC sublayer is responsible for buffer management and Automatic Repeat Request (ARQ). The MAC sublayer handles frame generation and performs Radio Resource Management (RRM), i.e. generates reservation messages and collects channel status information by scanning the available resources.
- The PHY layer hands through the packets from the RLC to the Radio Channel and vice versa, while it is able to generate a bit-error rate (BER) according to the measured signal-to-noise ratio (SNR) in the Radio Channel.
- The Radio Channel is responsible for the actual transmission of a packet, i.e. it buffers packets sent from the PHY, calculates resulting reception power and interference from other parallel transmissions and hands them back to all PHY instances of nodes in communication range, according to the propagation delay.

2.5.3 Performance Results in presence of Hidden Stations

For the following results the SDL-based simulation environment described in the previous chapter has been used. The results describe the behavior of the single frequency version of the FleetNet protocols in presence of hidden stations. For fully meshed networks the MAC scheme described in chapter 2.2 results in a system behavior of R-ALOHA without collisions, because in this case each node is in range of each other node and will overhear and respect neighboring nodes reservations. In presence of hidden stations, reservations of additional capacity may not be received by all nodes. Therefore, the information on the slot usage status of the system is not any longer unique in each node, and reservation conflicts (i.e. two nodes reserve the same slot for transmission) can occur, which may lead to packet collisions and will at least increase interference. The solution to the hidden-station problem has been described in section 2.2.4. The idea is to transmit information on used resources by means of the CSI (Channel State Indicator) table with each packet transmitted inside the CSBC. The CSI-table contains information on the receive status of all stations in its neighborhood. The performance of this approach is investigated in the next section.

Scenario and Constraints

In addition to the parameters described in section 4, some constraints are made for the event-driven simulations: The CSBCs for each node are predefined, i.e. all nodes start in the *Circuit Switched Mode* (cf. Fig. 2.4, Page 36). Each station has a limited buffering capacity of 100 packets inside the LLC protocol layer. The radio channel covers a simple pathloss model with $\gamma \approx 3$. This leads to a radio range of approx. 1000 m. Interference from other nodes using the same slot is taken into account resulting in packet collisions, if the Signal to Interference Ration (SIR) is below 7 dB. Two different scenarios are taken into account:

a) Scenario for Fully Meshed Network (**uniform**)
Since the radio range of the used air interface is assumed to be approx. 1000 m, the first scenario comprises a simulation area of 1000 m x 1000 m to simulate a fully meshed network, i.e. each node is in radio range of each other node. In total 20 nodes with a uniform distribution of their positions throughout the area is simulated. No node movement is considered. A single node is allowed to reserve up to 8 additional slots (besides its CSBC) for data transmission. With a maximum of 52 available slots inside one superframe (4 frames * 13 slots) and 20 slots reserved as CSBCs, 8 reserved data slots is the maximum amount to reserve the full capacity of the system for one single station. This scenario is called uniform in the sequel.
b) Scenarios with Hidden Stations (**hidden, acks, tables**)
To add hidden stations, a second scenario is observed. It triples the simulation area to 3000 m x 1000 m, where all nodes are distributed uniformly over the area (cf. Fig. 2.18).

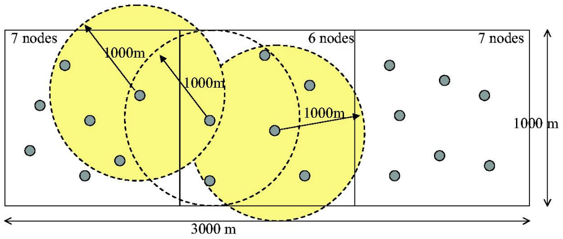


Fig. 2.18. Hidden station scenario with 3 groups (20 stations in total)

As the simulation area is tripled while the amount of nodes is kept constant, the achievable throughput in the system could be increased due to frequency-reuse. However, due to hidden stations reservation conflicts will occur and, therefore, packet collisions, which will reduce the maximum achievable throughput. In the basic scenario, called *hidden* in the sequel, no actions are performed if packet collisions are detected by the receiver: the sender continues sending until its buffer is empty. Collisions occur until one of the sending stations releases the conflicting resource. Since the sender is not aware of the collisions this simulation assumptions result in a worst-case behavior. For collision avoidance and resolution, two mechanisms were evaluated:

1. Usage of reservation acknowledgments:
   after transmitting a reservation of resources, the sender awaits an acknowledgment (ACK) message from its communication partner within one superframe (40 ms). If the receiver successfully decodes a data packet from the transmitter it will transmit the ACK in its next CSBC, otherwise it will send a negative ACK (NACK). A NACK is also transmitted if the receiver realizes the reserved resource to be undecodable or if it receives an additional reservation for the concerning resource. If the sender receives a NACK from its communication partner it will cancel the reservation. The ACK and NACK messages for multiple reservations can be combined into one message inside the CSBC.
2. Usage of CSI tables:
   Each station is periodically (in each CSBC) broadcasting a table, containing all resources used for receiving. The occurrence of a resource inside this table can be used as a reservation ACK by a reserving station; the absence of a reserved resource acts like a NACK.

The two scenarios are called *acks* and *tables* in the sequel.

Delay Performance

The following figures show the performance of the FleetNet air-interface for a population of 20 nodes. With 20 nodes inside an area of 3000 m x 1000 m, each node has about 4 direct neighbors and about 5 stations are hidden (cf. Fig. 2.18).

Fig. 2.19 shows the average message delay for a population of 20 nodes. The load axis is normalized to the achievable load in a 1000 m x 1000 m area with no hidden stations, i.e. to a network with full connectivity, respectively the max. achievable transmission rate of the air-interface. Each node has a maximum amount of 9 slots available for data transmission: one CSBC and up to 8 additional resources that can be reserved on demand. The amount of actually reserved resources depends on the buffer occupancy of each node. The first slot is reserved if more than 2 packets are stored. The second slot, if more than 5 packets are stored, because with one reserved slot 5 packets can be transmitted per superframe (one packet per frame and one packet inside the CSBC). The third slot is reserved if more than 10 packets are stored, and so on. As long as the traffic load is below saturation, the mean delay is almost constant and has a value defined by the waiting time until a slot for transmission is available. This is either the CSBC for reservation or one of the additionally reserved slots. The maximum supported load is

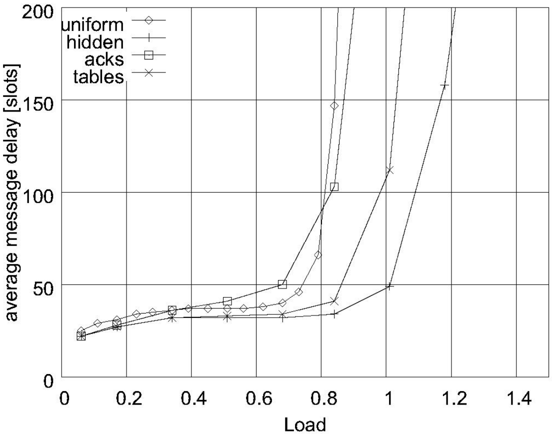


Fig. 2.19. Average message delay for 20 stations

approx. 85% in the uniform scenario, since ≈7% of the capacity is used for high priority services and the rest are either unused CSBCs or used CSBCs for signaling purposes. In the hidden scenario, the maximum supported load with reasonable delays increases to approx. 120% of the capacity. Using *tables* the max. supported load increases to 105%. If *acks* are used, a load of about 90% is reached. This increase, compared to the *uniform* approach, is caused by the increased simulation area and exploitation of frequency reuse, but the high collision rate due to hidden stations avoids a further increase. Nevertheless, it is worth mentioning that the supported load does not directly reflect the achievable throughput, since collisions are not taken into account in the delay curves and there is no ARQ scheme involved that guarantees the delivery of the offered load. The average message delay for low loads is even smaller than in the *uniform* scenario, because due to the tripled simulation area each node has fewer neighbors competing for the shared resources and more slots can be reserved by one station. To understand the differences between the simulated scenarios and to estimate the achievable throughput we have to take a look at the collision rate, depicted in Fig. 2.20. No collisions occur in the *uniform* scenario since reservation requests are correctly received and are accepted by all stations.

The very high collision rate of up to 50% in the *hidden* scenario can be reduced to about 20% in the cases *acks* or *tables* are used. (N)ACKs can be cumulated in one packet and will be transmitted by the receiver in its CSBC, i.e. once every superframe. If an (N)ACK packet is transmitted, no data packets can be sent inside the CSBC. In contrast, CSI tables can be transmitted piggybacked, i.e. the CSBC can be used to transmit data packets. This leads to a better delay performance using *tables*, i.e. the delay is lower and the max. achievable throughput is higher than using *acks*, because more resources are

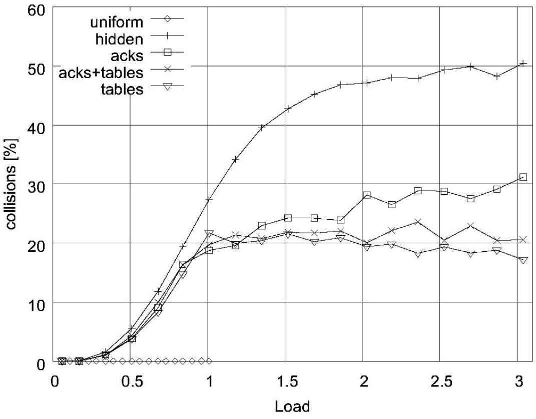


Fig. 2.20. Collision rate for 20 stations

available for data transmission. For low loads all conflict resolution approaches lead to comparable collision rates. With higher loads the *tables* approach gives the best results, because a sender can forecast free resources at the receiver - even if a sender is hidden - because all nodes broadcast their full reception status once every superframe. This is even more important if resources are scarce. If almost all resources are in use, the probability of situations increase where two stations reserve identical slots. Nevertheless, a collision rate of 20% remains, which is the result of the hidden stations and simultaneous reservation attempts of these station for the same resource. If the system is highly loaded there are only a few slots available for transmission. These slots will be reserved with a high probability by the hidden station and in the worst case the resolution of this conflict needs 4 frames. After the resolution the same problem might occur, i.e. the involved stations might select the same slot, if this is the only one available from the perspective of theses stations. This conflict can be partly resolved if other stations will repeat the resource requests (indicate in the CSI-table also those slots that will be reserved within the next superframe). Since this is only an incremental information the amount of signaling overhead is kept moderate and can be piggybacked like the CSI-table.

Fig. 2.21 shows the results for the overhead of the simulated scenarios. With increasing load the overhead increases to a maximum and then decreases again. At higher loads, the buffers of the transmitting stations are always filled, which leads to continuous packet trains. Resources are seldom released and less reservations are needed. The highest collision rate occurs for the *acks* approach because the CSBC is used for ACKs as well as for reservation messages. The *tables* approach leads to a significantly lower overhead, because CSI tables are transmitted piggybacked. Of course, both approaches lead to a

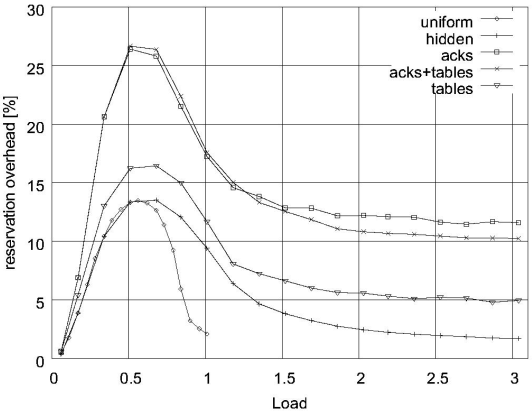


Fig. 2.21. Overhead for 20 stations

higher overhead than the pure *hidden* approach, where no conflict resolution methods are applied. Now, looking back at the delay performance, it seems that the highest load is supported with the *hidden* approach. But this is due to a collision rate that is not acceptable. That means that the throughput is lower than for the other approaches in presence of hidden stations. The support of high load conditions is the result of packet discarding. Those packets that collide will not be retransmitted. Only those packets that are successfully transferred will be taken into account for the delay. Consequently, the mean number of packets in the queue is very small for the *hidden* approach. Using CSI *tables* where all nodes report on resources they are receiving leads to the best results: the delay is the lowest, the collision rate is the lowest, and the max. achievable throughput is higher than for the *acks* approach.

2.5.4 Performance Results for dFDMA

In order to evaluate the performance of the proposed dFDMA concept two event-driven simulation environments have been built up: one based on the C++ class library CNCL[4] used for the evaluation of the general protocol behavior and one SDL based simulation environment, used for evaluating the resource allocation problems depicted in Sec. 2.4.3.

Scenario and Constraints

Different to the scenario described in section 4, only a population of M = 18 stations is simulated, because this gives a uniform distribution of nodes on the

[4] CNCL: Communication Networks Class Library, available from http://www.comnets.rwth-aachen.de

three frequency patterns. In the single-frequency approach (no dFDMA) one slot for the CSBC is provided for each station in every superframe, whereas a superframe contains $n_{SF} = 4$ frames. Opposed to this for the dFDMA approach two CSBC slots are reserved for each station in every superframe. Data packets can be served in CSBC slots. If this capacity is not sufficient, it is assumed that another slot can be reserved by means of an inband- signaling message, which is transmitted piggybacked with the user data packet. This additionally reserved slot is then used in every frame in the case no dFDMA is applied, whereas the slot is used in every superframe in case of dFDMA. For dFDMA the stations are uniformly distributed on the three frequency patterns. For the sum of 18 stations there are always 6 stations assigned to each individual frequency pattern. Each station is assigned one target destination, which is not changed during the simulation. Two different allocation schemes for the destinations exist:

1. **diff**: Source and destination belong to different frequency-patterns (except on the common coordination frequency they only can communicate during frame 2, where they can exploit all possible frequencies).
2. **com**: Source and destination belong to one common, i.e. the same frequency-pattern.

Within the simulations all stations belong to the same allocation scheme, either *diff* or *com*.

Delay Performance

The resulting mean delay as function of the load normalized to the max. possible capacity of one frequency channel for the C++-simulation with the CNCL is depicted in Fig. 2.22. Under low load the delay for dFDMA using different frequency patterns (*dFDMA/diff*) is comparable to the delay when not using dFDMA since the one CSBC slot every superframe is sufficient to serve the user data. For *dFDMA/diff* only one of the two CSBC slots a station has reserved can be exploited for data exchange since different frequency patterns have only one common EX-phase comprising the CSBCs. In contrast to this stations using common frequency patterns for data transfer (*dFDMA/com*) utilize both CSBC and can considerably reduce the delay under low load conditions. In mean only half of the delay is needed for *dFDMA/com* since twice the capacity is provided, i.e. the mean delay is reduced approximately from half of a superframe to a quarter of a superframe. However, with increasing load the delay dramatically increases for dFDMA, especially for *dFDMA/diff*, since the additional capacity provided by one slot every superframe is far too small.

For *dFDMA/com* the additional CSBC utilized for data transfer reduces this deficiency, but still, the delay considerably increases. However, compared to the single- frequency appliance it can be recognized that *dFDMA/com* can offer higher total throughput. This is the result of the other available

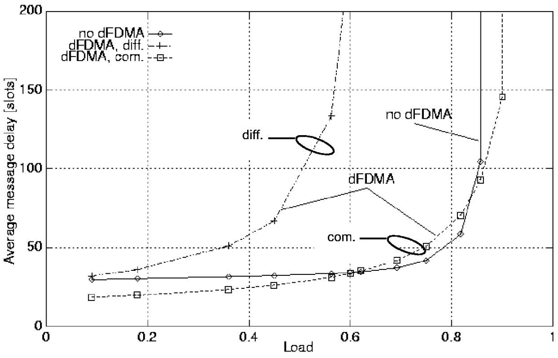


Fig. 2.22. Mean delay using dFDMA

frequencies that can be used to reserve always an additional slot for data transfer. Therefore, each station under saturation for *dFDMA/com* has two CSBCs and another slot every superframe for data transfer. This reduces to one CSBC and one additional slot for *dFDMA/diff*. When using no dFDMA all stations have to share the nSF * N = 52 slots in a superframe for data transfer. From these slots further M = 18 slots have to be subtracted, which are reserved for the CSBC. Hence, each station has one CSBC and in mean another 34/18 slots in every superframe, which is less than for *dFDMA/com*.

Increasing Transmit Capacity

Nevertheless, such low utilization of three available frequencies in case of dFDMA is far below as been expected from three times the capacity of a single- frequency. The main reason for the low throughput for dFDMA is the limitation of reserved capacity to one slot only in every superframe besides the capacity provided by CSBCs. In the following it is therefore assumed that each station is allowed to reserve up to 10 slots on all frequencies to exploit the available resources in an efficient way. To reserve another slot the number of packets in the queue is taken into account. If the number exceeds a given threshold multiplied by the number of already reserved slots another slot is reserved. The first slot is reserved when the CSBC slot is available and if more than two packets are in the queue. For a threshold of 3 the second and third slot is reserved when more than 3, respectively 6 packets are in the queue. Two different approaches to increase the capacity are investigated: 1. The idealized approach, where no additional signaling-overhead exists, and 2. the more realistic approach, in which a reservation request for further capacity requires one slot. In the idealized case, which has been presented in the previous section, it is assumed that this reservation message is transmitted piggybacked with the user data packet. In this case the reservation of further

slots does not generate any signaling overhead. Moreover, a new slot can be reserved piggybacked with a data packet that is transmitted on an arbitrary frequency at any position in the frame. Though this piggyback reservation will be received by the destination and can be used to increase the capacity on an existing link, it might not be overheard by other stations and, hence, is a source for a potential reservation conflict. In the more realistic approach it is assumed that only the CSBC can be used to increase transmission capacity and at the same time no user data packet can be served. The additionally reserved slots are used in every superframe as long as packets are in the queue. After all packets have been transmitted, all temporarily reserved slots are released with one single message transmitted with the last user data packet.
The resulting mean delay as function of the load normalized to the maximum possible capacity of one frequency channel is depicted in Fig. 2.23. All stations use a common frequency-pattern (*com*).

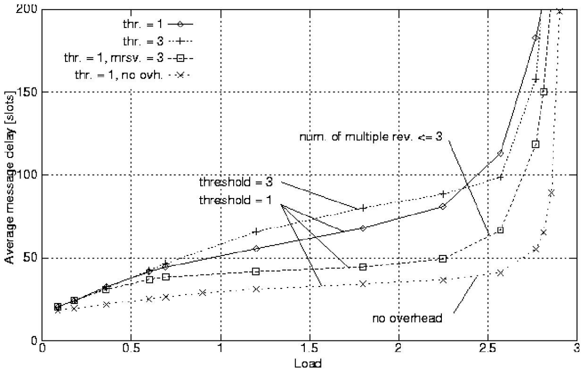


Fig. 2.23. Mean delay for dFDMA with and without considering signaling overhead

As indicated in Fig. 2.23 the delay decreases with smaller thresholds (thr. = 1 instead of thr. = 3), as expected, since more resources are reserved to emptying the queue. At this point it has to be mentioned that with smaller thresholds the signaling overhead for reservation requests increase. It can be further recognized that with *dFDMA/com* and an adaptive and variable number of reserved slots the throughput can be increased close to the theoretical maximum of three times the value a single-frequency provides. The curves with the higher delays result from simulations taking into account the signaling overhead. For a load of 200% the delay is approx. 70 and 80 slots for a threshold of 1, respectively 3, which corresponds to a delay of 50, respectively 57 ms. Compared to the delay of 30 slots for the idealized approach without overhead (no overhead), the impact of signaling is considerable. Because of the signaling messages, which are only allowed to be transmitted on the CSBC, the available capacity for signaling requests is reduced in contrast to the idealized approach where the CSBC and all reserved slots for a connection can be utilized. Hence, it takes much more time in the realistic approach to reserve new slots on demand. In addition, the available transmit capacity is reduced by the signaling capacity needed to transfer the requests. This becomes more problematic for higher loads close to saturation, e.g., above 240% the delay for the low threshold of 1 results in higher delays than for a threshold of 3. However, for lower loads there is enough capacity left for signaling purposes and a lower threshold results in a better performance. Still, the time to reserve new slots is comparably large. To increase the transmission capacity faster an extension to the common approach is introduced. Instead of reserving only one slot with a reservation request it is now allowed to reserve up to three slots simultaneously, if the number of packets in the queue indicates such needs. With this multiple-slot- reservation (MRSV) it becomes possible to reserve 3 additional slots on each CSBC, which makes up to 6 slots every superframe, if the destination uses the same frequency-pattern. Therefore, in two superframes the maximum allowed number of 10 slots can be made available to serve temporary peak traffic, e.g., batch-arrivals of user data. The resulting delay for MRSV is represented by the curve denoted by ”num. of multiple resv. $\leq$ 3” in Fig. 2.23. As can be seen in this figure the delay can be decreased by more than 20 slots for a load of 200% and a threshold of 1. The resulting delay comes close to the delay for the idealized simulation assumptions. The additional delay for the realistic approach is less than one frame of 10 ms.

Resource Allocation Problems

In this section a simulative investigation of the described resource allocation problems and the proposed solutions (cf. Sec. 2.4.3) is performed and the results are presented. The simulation environment does not perform any link layer acknowledgments or retransmissions, i.e. if collisions occur or transmissions cannot be received on a channel because the receiver is currently working on a different frequency the packet is lost and the transmitter is not informed about this event, like in the unacknowledged mode of UTRA TDD. The following graphs represent the results of 4 simulated scenarios:

1. **central**: the system is equipped with a central reservation table, i.e. all stations are fully informed about the slot status;
2. **decentral**: all stations have their own reservation table; they gather the information only by overhearing reservations and ongoing data transmissions;
3. **repetition**: like decentral, but for reservations regarding slots in frame 2 of the superframe the protocol performs a repetition of reservations in the next CSBC;
4. **tables**: like decentral, but each station transmits a reservation table with each packet on the CSBC, where all slots are listed the station is receiv-

ing on in the next superframe. This corresponds to the Channel State Information (CSI)-table to combat hidden stations.

First we have a look on the resulting packet loss as function of the load normalized to the maximum possible capacity of one frequency channel (cf. Fig. 2.24). In the central scenario no packets are lost (we assume no channel errors) and no reservation conflicts occur because the stations are fully informed about the channel allocations. Due to the depicted problems of reservation conflicts and irresolvable reservations, the packet loss increases in the other scenarios.

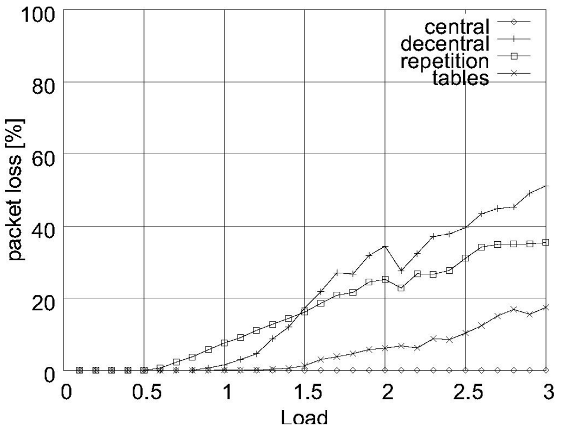


Fig. 2.24. Packet loss in dFDMA

The *decentral* approach performs worst, because no conflict resolution methods are applied. That means, if packets collide because of reservation conflicts, collisions will also occur in the slots in the following frames until one of the stations will release the slot because it has no packets to transmit. The spreading of reservation *tables* performs better than *repetition* of reservations. The reason becomes clearer, if we look at the two possibilities of packet losses: collisions and irresolvable reservations, which are depicted in Fig. 2.25 and Fig. 2.26. In the *decentral* scenario packet loss is mainly caused by collisions due to reservation conflicts. These collisions can be combated by *repetition* of reservation messages and by spreading of reservation *tables.* The repetition significantly reduces the collision rate, since reservation conflicts will be recognized and the stations will stop to use the respective slot. But the spreading of reservation tables performs better. This is because the repetition is sender-based, but collisions happen at the receiver. If the receiver resolves the conflict by sending a table where it successfully receives on, conflicts can be resolved earlier. The rest of the packet loss is caused by irresolvable reservations, depicted in Fig. 2.26. The *repetition* of reservation messages does not reduce the number of irresolvable reservations, they are increased instead. If reservations are repeated more stations regard them and may have the wrong information if the reservation is canceled afterwards because the receiver is tuned to a different frequency. Even the spreading of reservation *tables* does not fully combat this problem, because between two table transmissions the situation can change due to further reservations of other stations or own reservation cancellations. This information may not be available at the sender.

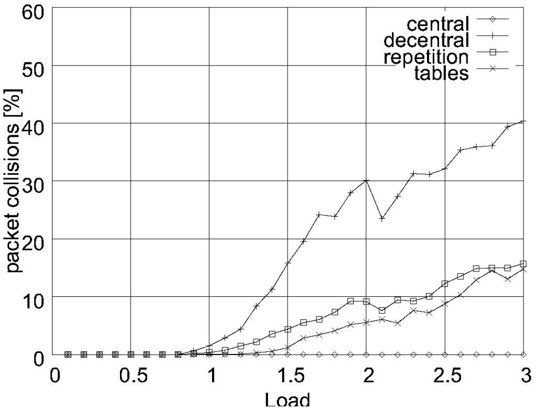


Fig. 2.25. Packet collisions in dFDMA

In an enhanced version of the protocol, where ARQ is introduced, reservation conflicts and irresolvable reservations will be further combated by acknowledgments and link layer retransmissions. This will reduce the packet losses but increase the mean message delay. Additionally, reservation tables can be added providing information on which channels a sender is transmitting. This will have the same effect like a repetition of reservations but - if the whole reservation still fits into one CSBC - without occupying a whole additional slot. In Fig. 2.27 the resulting mean message delay as function of the load normalized to the maximum possible capacity of one frequency channel is shown. Compared to the *central* scenario, the mean delay increases because of the additional overhead in the protocol. Here the *repetition* of reservation performs worst, because more capacity is used for the repetition, which can be used for data transmission in the other scenarios. Due to performed reservation cancellations the delay slightly increases in the case of the *table* scenario. The maximum supported load of the *repetition* and *tables* scenario is comparable to the *central* scenario. In the *decentral* scenario the highest load values are supported, but this is due to the fact that only transmitted packets are counted for the statistical calculation. If more packets are lost, the throughput

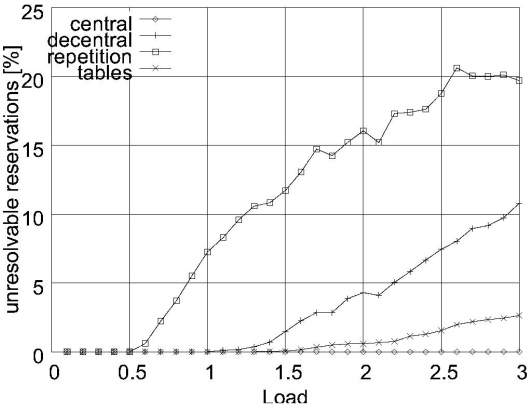


Fig. 2.26. Irresolvable reservations in dFDMA

is smaller than the offered load and packets with high delay will be dropped. This improves the delay performance close to saturation of the system on cost of increased packet losses. And the overall throughput decreases with increasing number of collisions.

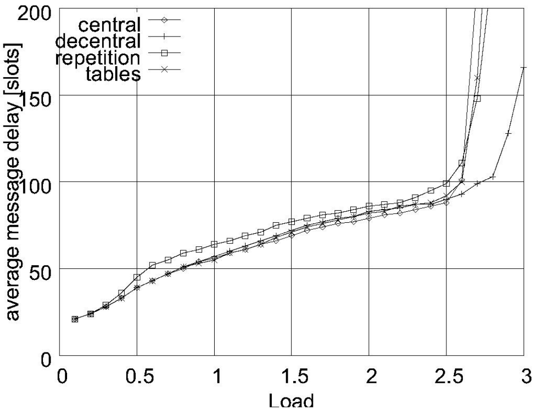


Fig. 2.27. Mean delay in dFDMA

Automatic Repeat Request

An SDL based simulation environment has been set up for the purpose of validating the analytically obtained results. A frame with duration of T = 10 ms comprising N = 14 slots is considered. No velocity is assumed. Uncorrelated channel errors are considered, but as a first approximation the reverse channel is supposed to be error free. No capture-effect is taken into account. One packet, either data or ARQ can be served within one channel. The performance has not only been tested in a single user scenario but as well in a multi user scenario where the network is modeled with a fully meshed topology and a population of M = 8 stations. All stations have identical message arrival statistics that follow a stationary Poisson process with rate λ. Each station has a limited buffering capacity of 100 packets. One channel for the CSBC is provided for each station in every superframe, i.e. 4 TDD frames. The multi user scenario has been separated in two cases as well: a) where each user can reserve a maximum of one slot every frame for data transmission, and b) where each user is allowed to reserve up to 3 slots / frame. We first take a look on the case a). The obtained overhead efficiency is in general lower than in the analytical case, but shows similar behavior. Both protocols show very similar overhead efficiency for the expected packet error rates, and the gain of SR over GBN decreases with lower PER. However, the gain of SR over GBN is much smaller than for the analytical evaluations and for PER below $10e^{-3}$ there is no recognizable difference between GBN and SR (see Fig. 2.28). Notice, that for high PER, e.g. 10^{-1} the improvement of SR over GBN grows with the size of the window. In the SR scheme the number of ARQ messages sent does not depend on the number of erroneously received packets but only on the window sizes, therefore with larger window sizes the number of ARQ messages that have to be sent decreases. In the GBN scheme the number of ARQ messages depends on the window sizes but also on the number of erroneously received packets. For window sizes ≥ 64 the number of GBN ARQ messages remains constant and in the same order of magnitude as the number of erroneously received packets, while the number of SR ARQ messages is halved each time the window size grows one step.

In the multi user case b) every user is allowed to reserve up to 3 slots per frame for its transmissions. The results in Fig. 2.29 are, however, quite similar to the results obtained in case a). The reason for that results is that in average each station occupies less than one slot per frame, much fewer than the maximum of 3 slots allowed. In fact, separate from the number of slots per frame that a station is allowed to reserve, in average a station reserves actually only one or two slots per frame if it has data to transmit. In many cases no data have to be transmitted and the station reserves no slot at all. The difference in the overhead efficiency between SR and GBN is much smaller than estimated in the analytical derivations for a single user occupying the whole frame because the number of packets that are retransmitted in case of an erroneous packet is significantly smaller in the simulations.

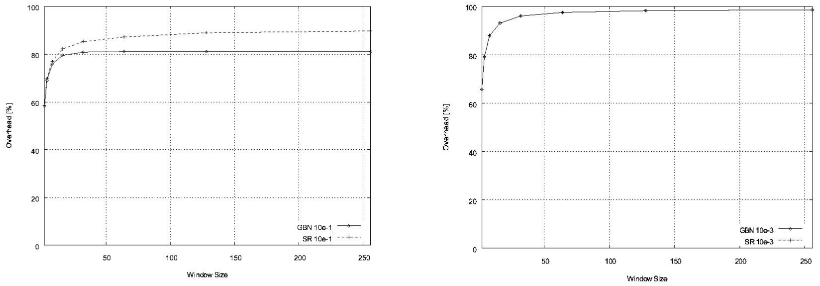


Fig. 2.28. Multi user case a) PER=10e-1 (left) PER=10e-3 (right).

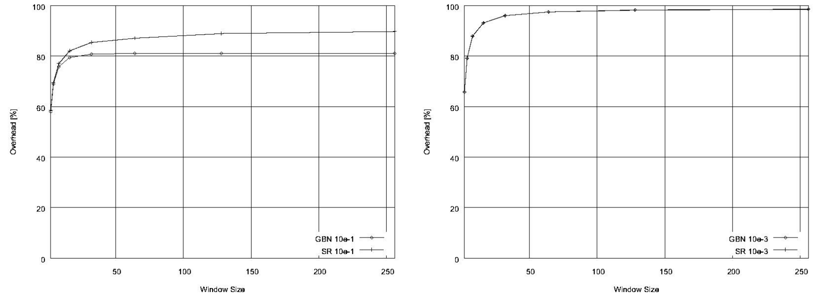


Fig. 2.29. Multi user case b) PER=10e-1 (left) PER=10e-3 (right).

Delay with ARQ In this work we consider the average message delay at Logical Link Control (LLC) level, i.e. between the LLC layers of two peer stations. The average message delay comprises then two terms: the average access delay and the average transmission delay. The average access delay refers to the mean time a packet must wait in the LLC input buffer before a free slot is available for transmission. The mean duration of a successful packet transmission between two LLC peer layers correspond to the average transmission delay, esp. the delay caused by retransmissions. GbN ARQ accepts only error free packets received in sequence. SR ARQ, however, accepts also error free packets that are not in sequence. Therefore, if we considered the average message delay at application layer, another delay term would have to be accounted for the SR scheme, i.e. the average re-ordering delay at the reception buffers. Only the multi user scenario has been simulated. In the case b) the mean message delay has been derived. The mean delay expected when no errors occurs in the channel was calculated in section 2.5.1, p. 51, and is used in Fig. 2.30 and Fig. 2.31 as a reference. As long as the traffic load is below saturation the delay is almost constant and takes on a value of approx. 35 slots.

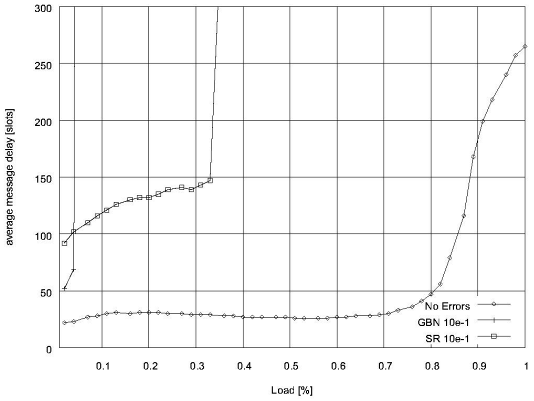


Fig. 2.30. Average message delay, $PER = 10e^{-1}$.

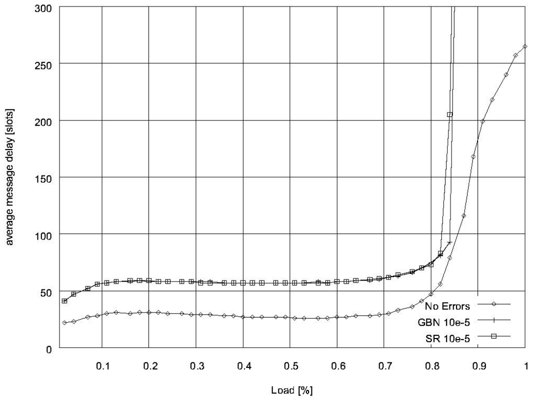


Fig. 2.31. Average message delay, $PER = 10e^{-5}$.

As expected, with large PER the mean delay highly increases; many packets are erroneously received and must be retransmitted. Under large PER, SR shows a better performance both with regard to overhead (cf. Fig. 2.28 and Fig. 2.29) and delay (cf. Fig. 2.30 and Fig. 2.31) than GBN. GBN reaches the saturation much earlier than SR. The reason is, as ARQ messages and reservation packets are both sent within the CSBC, if two many ARQ messages have to be sent (as in GBN), the rate of sent reservation packets decreases and therefore not enough new resources can be reserved and the input buffers grow fast until saturation. As well, commonly an erroneous packet originates more retransmissions in the GBN case than in the SR case. While the sender is busy sending retransmissions, the input buffer is growing up all the time but new packets can not be sent until all retransmissions have been completed. The performance of both ARQ protocols with regard to the delay has a similar evolution as with regard to overhead efficiency. Fig. 2.31 shows that for lower PER, below $10e^{-3}$, the mean delay for both ARQ protocols is very similar. The saturation state is reached with both protocols almost at the same traffic load than in the no error case. The delay of aprox. 60 slots is almost constant until saturation.

Proposal of an ARQ scheme for UTRA TDD ad hoc mode We propose the GBN protocol as the ARQ mechanism most suitable for FleetNet. We have shown that its efficiency in terms of overhead and average message delay is comparable to the one of SR for the expected error rates. It offers more flexibility due to its smaller signaling overhead over the reverse channel; it has the advantage that a GBN ACK/NACK could always be sent inside the CSBC and as well be piggybacked within a data packet. Our simulation environment will be enhanced with a piggybacking facility, and we expect that with these modification the efficiency of GBN will exceed that of SR for PER below 10^{-3}. Beyond, as it is supposed that stations usually reserve only a few slots per frame, after an erroneous packet only few packets must be retransmitted. The hardware implementation is also simpler, most notably at the receiver, where no reception buffers are required to intermediate store correctly received out of sequence packets. Moreover, with the SR scheme another delay term should have to be summed up, i.e. the re-ordering delay at the receiver. The model of the implemented RLC layer for FleetNet, using GBN as the ARQ scheme and the buffer management is illustrated with help of the following Fig. 2.32.

2.5.5 Impact of Mobility

Major services supported by FleetNet will be road traffic telematics and mission critical services like emergency notifications and services for co-operative driver assistance, which put very high demands on the air-interface and the used protocols. High relative velocities up to 500 km/h between oncoming vehicles in a highway scenario will lead to frequent topology changes, i.e. a very high network dynamic. Generally, it has to be answered, if the developed dynamic reservation scheme is reasonable for VANETs and its high relative velocities. A significant parameter for medium access in mobile ad hoc networks is the grade of topology changes. Topology dynamics will have impact on various parts of the protocols. The mean possible communication duration between vehicles will have impact on the amount of reservation mes-

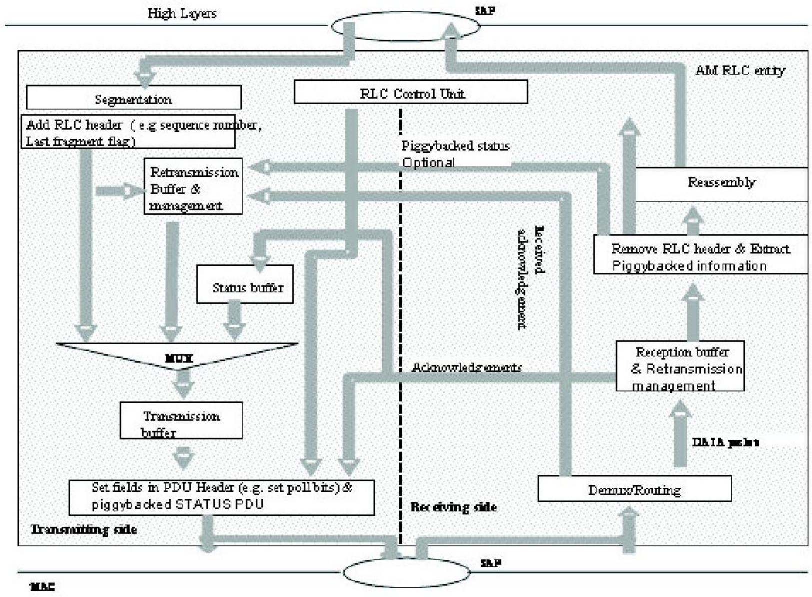

High Layers
SAP
AM RLC entity
Segmentation
RLC Control Unit
Add RLC header (e.g sequence number, Last fragment flag)
Piggybacked status Optional
Retransmission Buffer & management
Reassembly
Received acknowledgement
Remove RLC header & Extract Piggybacked information
Status buffer
MUX
Acknowledgements
Reception buffer & Retransmission management
Transmission buffer
DATA packet
Set fields in PDU Header (e.g. set poll bits) & piggybacked STATUS PDU
Demux/Routing
Receiving side
Transmitting side
SAP
MAC

Fig. 2.32. Model of an RLC layer for the GBN ARQ scheme

sages transmitted. Furthermore, the grade of topology changes will affect the amount of reservation conflicts and collisions inside the system, if vehicles approach each other, which have reserved identical resources. Simulation of traffic scenarios and statistical investigations on the traffic dynamics based on classical vehicular traffic theory, e.g. [33], shall give realistic estimations on possible communication durations and speed of topology changes. In the classical vehicular traffic theory (e.g. [34], [33]) freeway traffic is described by three elementary parameters: traffic density ρ_{veh} in [veh/km], traffic flow q in [veh/s] and net time gap τ in [s]. These quantities can be related together by their average values [35] as shown in Equations (18)-(20). Herein l_m is the average (mean) length of vehicles, d_m the average distance between vehicles and v_m the average speed in [m/s] of vehicles and ρ_{veh} the traffic density on the considered freeway section:

$$d_m = \frac{1000}{\rho_{veh}} - l_m; \tag{2.22}$$

$$\tau_m = \frac{d_m}{v_m} \tag{2.23}$$

$$q_m = \frac{1}{\tau_m} \tag{2.24}$$

In Equations (19) and (20) the mean time gap τ_m and mean traffic flow q_m are calculated for given values of v_m and ρ_{veh}. Otherwise, if a minimum time gap τ_{min} and a traffic density ρ_{veh} are given, one can calculate a velocity parameter v_p. This value can be interpreted as a parameter of the route and it represents a possible maximum velocity on this route. Real average velocities, the so-called average free velocities $v_{m,free}$, are always below or equal to this limit, cf. Eq.(21):

$$v_{m,free} \leq v_p = \frac{d_m}{\tau_{min}} \tag{2.25}$$

If traffic density is low, vehicles are assumed to drive with this free velocity $v_{m,free}$. Additional vehicles do not diminish the average driving velocity $v_{m,free}$, but only v_p. Thus, additional vehicles result in an increase of the traffic flow q and shorter time gaps τ_m. This traffic state is called undisturbed traffic. If traffic density becomes higher so that it is not longer possible to drive by $v_{m,free}$, the driven velocity will reduce to v_p and the traffic state is called disturbed traffic. After introducing these vehicular traffic theory fundamentals we want to have a look on the statistical distribution of the velocity, which can be described by the random variable v. The velocity is generally assumed to be normal distributed [34]. Therefore, to the probability density function (pdf) and the probability distribution function (PDF) of velocity applies:

$$p_v = \frac{1}{\sigma\sqrt{2\pi}}\, e^{\frac{(v-\mu)^2}{2\sigma^2}} \tag{2.26}$$

$$P(v \leq V) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^V e^{\frac{(v-\mu)^2}{2\sigma^2}} dv \tag{2.27}$$

whereby μ and σ^2 are average value and variance of velocity according to the usual notation. The following two basic communication scenarios are considered [36]: a) all vehicles move in the same direction and b) oncoming traffic. For the different scenarios the possible communication durations between two vehicles are calculated. This gives a rough indication if the topology is approximately stable during a period of time that is long enough to make resource reservations reasonable. A vehicle can either be within detection or communication radius of another vehicle. The detection radius defines the area where a transmission of any station can be detected, whereas the communication radius defines the area where a signal can be decoded with high probability. We first consider an undisturbed traffic scenario. Velocities of vehicles are constant, whereby the value of velocity is generally assumed as Gaussian distributed.

a) All vehicles are driving in the same direction:

From the Gaussian distribution using Eq. (22) the probability density function (pdf) $p_t(t)$ of communication duration can be calculated as [37]:

$$p_t(t) = \frac{2R}{\sigma_{\Delta v}\sqrt{2\pi}} \frac{1}{t^2} e^{\frac{(2R/T-\mu_{\Delta v})^2}{2\sigma_{\Delta v}{}^2}} \quad \text{for} t \geq 0 \tag{2.28}$$

wherein R is the communication radius, and $\mu_{\Delta v}$ and ${\sigma_{\Delta v}}^2$ are mean and variance of relative speed. Integrating Eq. (24) we get the probability distribution function (PDF) $P_t(t \leq T)$ of the communication duration, which is shown in Figure 2.33 for average velocities between 30 km/h and 150 km/h and a communication radius of $R = 1$ km. Typical durations exceed 280 s for a relative speed of 30 km/h and 55 s for 150 km/h in 95% of the cases if vehicles are driving in the same direction.

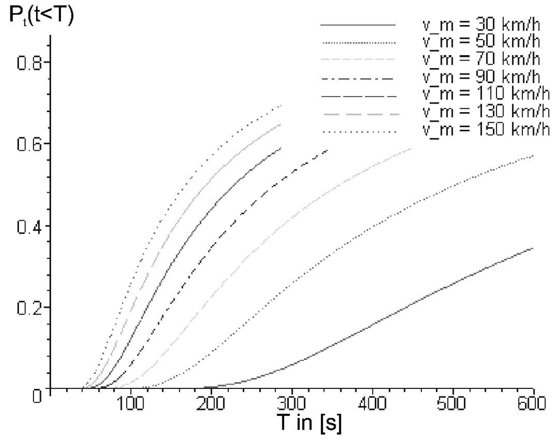


Fig. 2.33. PDF of communication duration t_{comm}

b) Oncoming traffic:

For this scenario a similar equation to Eq. (24) can be inferred. From this we obtain the PDFs shown in Figure 2.34. With increasing speed the communication duration decreases as expected. Typical durations exceed 140 s for a relative speed of 30 km/h and about 30 s for 150 km/h in 95% of the cases if oncoming traffic is considered. Therefore, even for a mean velocity of 150 km/h there is a high probability to have communication periods with more than 3000 MAC-frames, which makes reservations reasonable for our simple scenario.

Simulation results presented in [37] show, that Eq. 24 can give a rough indication for possible communication durations between two vehicles. Although communication durations are shorter in real life scenarios than in theory, it has to be emphasized, that in communication scenarios, where the focus is basically on vehicles that are driving in the same direction, the situation is much more relaxed. Running an ad hoc network in an inter vehicle communication scenario seems to be very challenging, regarding the very high dynamics and frequent topology changes. However, the use of UTRA TDD as the air interface for FleetNet gives us enough flexibility to handle these dynamics. The

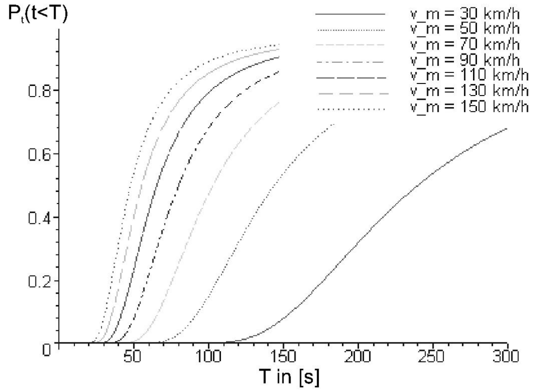


Fig. 2.34. PDF of communication duration $t_{comm,on}$

presented calculations and simulations show that MAC and RRM scheme proposed for VANETs can handle the requirements that come from realistic traffic scenarios in freeway environments as the communication durations between arbitrary stations typically exceeds 3000 MAC frames. Consequently, reservation based schemes as proposed for UTRA TDD ad hoc are feasible.

2.6 Conclusions

The focus of this chapter was the adaptation of UMTS UTRA TDD to a VANET setting. The proposed solution allows for communication distances of more than 1 km and very high relative velocities. Both properties are of significant importance in VANETs.

To adapt the cellular-based UMTS UTRA TDD to the requirements of a VANET, changes on the data link control layer were proposed. These changes comprised a reservation-based single-transmitter algorithm for the medium access control. In addition, a distributed radio resources management scheme based on the dissemination of receiver-based status information has been presented. Theoretical analysis as well as event-driven simulations indicate high values for the throughput and almost constant delays until network saturation.

It is expected that this new approach for an air-interface is not only well suited for inter-vehicle communication, but also for other self-organizing wireless networks that need to support high communication ranges and high mobility.

References

1. Andrisano, O., e.a.: Intelligent transportation systems: The role of third-generation mobile radio networks. IEEE Communications Magazine **vol. 38, no. 9** (2000)
2. Walke, B.: Mobile Radio Networks. Wiley & Sons (1999)
3. Hartenstein, H., Bochow, B., Ebner, A., Lott, M., Radimirsch, M., Vollmer, D.: Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project. Proc. of the ACM MobiHoc'01 (2001) 259–262
4. DSRC: (Dedicated short range communications (dsrc) home)
5. M. Haardt, e.a.: Td-cdma based utra tdd mode. IEEE Journal on Selected Areas in Communications **18, no. 8** (2000) 1375–1385
6. Lott, M., Halfmann, R., Schulz, E., Radimirsch, M.: Medium access and radio resource management for ad hoc networks based on utra tdd. In Proc. of MobiHoc 2001 (2001)
7. CarTALK. (http://www.cartalk2000.net)
8. 3GPP: Physical channels and mapping of transport channels onto physical channels (tdd) (2001)
9. 3GPP: Spreading and modulation (tdd) (2002)
10. DRIVE. (http://www.ist-drive.org)
11. 3GPP: Opportunity driven multiple access (1999)
12. Kim, Y., Nakagawa, M.: R-aloha protocol for ss inter-vehicle communication network using head spacing information. IEICE Trans. Commun. **Vol. E79-B, No.9** (1996) 1309–1315
13. Lott, M.: Random access for wireless ad hoc broadband networks. Proc. European Wireless (EW'99) (1999) 421–426
14. Zhu, W., Hellmich, T., Walke, B.: Dcap, a decentral channel access protocol: performance analysis. (1991) 463–468
15. Coletti, L., Riato, N., Capone, A., Fratta, L.: Architectural and technical aspects for ad hoc networks based on utra tdd for inter-vehicle communication. Proc. IST Mobile and Wireless Communications Summit (2003)
16. Borgonovo, F., Capone, A., Cesana, M., Fratta, L.: Ad hoc mac: a new flexible and reliable mac architectre for ad hoc networks. Proc. IEEE Wireless Communications and Networking Conference (2003)
17. Coletti, L., Moretti, L., Riato, N., Borgonovo, F., Capone, A., Cesana, M., Fratta, L.: Inter-vehicle communication: a new frontier of adhoc networking. Proc. Mediterranean Ad hoc Networking Workshop MED-HOC NET (2003)
18. Guirao, M.P., Meinke, M., Lott, M., Jobmann, K.: Automatic repeat request in ad hoc networks based on utra tdd. Proc. World Wireless Congress WWC04 (2004)
19. Lott, M., Walke, B.: A wireless ad hoc multihop broadband network with quality of service support. Information Systems Technology Panel Symposium on Tactical Mobile Communication (TMC'99) (1999)
20. Lott, M., Walke, B.: Performance analysis of a wireless ad hoc network with qos support. Telecommunication Systems **vol, 16, no.1-2** (2001) 115–134
21. Meincke, M., Lott, M., Jobmann, K.: Reservation conflicts in a novel air interface for ad hoc networks based on utra tdd. Proc. of IEEE VTC'03 Fall, (2003)
22. Borgonovo, F., Capone, A., Cesana, M., Fratta, L.: Adhoc mac: a new mac architecture for ad hoc networks providing efficient and reliable point-to-point

and broadcast services. ACM Wireless Networks (WINET) **Vol. 10, Issue 4** (2004) 359–366

23. Bossert, M.: Channel Coding for Telecommunications. John Wiley and Sons Ltd (1999)
24. Stallings, W.: Data & Computer Communications. Prentice Hall, New Jersey (2000)
25. Kurosse, J., Ross, K.: Computernetze. Pearson Studium (2002)
26. Bertsekas, D., Gallager, R.: Data Networks. Prentice Hall, New York (1992)
27. Tanenbaum, A.: Computer Networks. Prentice Hall, Munich (1997)
28. 3GPP: Rlc protocol specification (2001)
29. Lott, M., Halfmann, R., Meincke, M.: A frequency agile air-interface for inter-vehicle communication. Proc. ICT 2003 (2003)
30. Lam, S.: Packet broadcast networks - a performance analysis of the r-aloha protocol. IEEE Trans. on Computers **vol. C-29, no. 7** (1980) 596–603.
31. 3GPP: Universal mobile telecommunications system (umts); selection procedures for the choice of radio transmission technologies of the umts (1998)
32. Schnabel, W., Lohse, D.: Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung. Verlag für Bauwesen, Berlin, Bd. 1, 2 Auflage (1997)
33. Leutzbach, W.: Introduction to the Theory of Traffic Flow. Springer, Berlin, (1988)
34. Vollmer, D., Hiller, A.: Problemorientierte verkehrsmodellierung auf bundesautobahnen (2001)
35. CNCL: (Cncl: Communication networks class library)
36. Rudack, M., Meincke, M., Lott, M.: On the dynamics of ad hoc networks for inter-vehicle communications (ivc). Proc. of Int´l Conf. on Wireless Networks (ICWN´02) (2002)
37. Rudack, M., Meincke, M., Lott, M., Jobmann, K.: On traffic dynamical aspects of inter vehicle communications (ivc). IEEE VTC‘03 Fall (2003)

3

Forwarding of Emergency Notifications in One-dimensional Networks

Peter Tondl and María Dolores Pérez Guirao

University of Hannover, Institute of Communications Engineering
Appelstr. 9A, 30167 Hannover, Germany
{tondl,perez}@ant.uni-hannover.de

Summary. Inter-Vehicle Communication introduces some challenges to the standard performance of ad-hoc networks. In particular, high speeds and one-dimensionality of road scenarios are different to other ad-hoc communication scenarios. One key service that will make use of inter-vehicle communications is the dissemination of safety related messages, so called Emergency Notifications. Such messages are only released in case of an emergency when an immediate and, potentially, automatic reaction of the surrounding vehicles is required to prevent critical situations and further accidents. This chapter investigates forwarding strategies proposed for the distribution of safety related messages within these networks.

3.1 Introduction

Driving a car is one of the most dangerous human activities. The innovation progress of car engineering has contributed in the last decades to proportionate a high standard of passive safety and comfort in modern cars.

Passive safety systems can ensure the survival of the driver in traffic accidents only in case of low velocities. Intelligent Transportation Systems (ITS) have been proposed with the aim of using advanced technologies to improve safety and efficiency of transportation systems.

One of the main features of such systems is the use of wireless communication among vehicles, called Inter-Vehicle Communication (IVC). Floating car data such as speed, acceleration and braking conditions are transmitted. Human visual attention is very limited and a driver has only incomplete knowledge about the exact positions and velocities of surrounding vehicles. Thus, accidents, congestions and other traffic associated problems are the consequence of the inability of drivers to evaluate complex traffic situations correctly and instantly. Reactions to dangerous situations using Inter-Vehicle Communication are predicted to be faster and more reliable than human reactions, since advance warning is given, and actions to avoid accidents can be taken.

This chapter investigates forwarding strategies proposed for the distribution of safety related messages as a part of an Intelligent Transportation System.

3.2 Multi-hop Forwarding Strategies

Emergency Notifications show high requirements with respect to end-to-end delay but have small bandwidth requirements. Since the functionality of Emergency Notifications depends largely upon the reliability and delay of their data transmission, an instant and exclusive access to the shared physical medium is crucial. For that reason, a predefined part of the channel may be reserved for the data transmission of Emergency Notifications only. The goal of this section is the development of a forwarding strategy for the dissemination of these Emergency Notifications.

3.2.1 Challenges of Message Propagation

Challenges of message propagations regarding safety related messages like Emergency Notifications depend on their underlying scenarios. In the following we have chosen a so called "worst case scenario" which means there was only one message sent by a car in case of an accident of that car. As we can see later a different approach makes no sense for investigations described in this section. However, each vehicle with ad hoc network capabilities should be reached by a notification just demanding a minimal count of repetitions. Furthermore, despite having no infrastructure of any kind information about an accident should remain in its geographical area over a certain time. The aim of these requirements is to disseminate information about the accident quickly and efficiently to any vehicle affected by the dangerous situation.

In case of first receiving an Emergency Notification a vehicle has to wait a given time with its repetition of the message, based on the algorithm described in Sect. 3.2.3. In addition to this time the vehicle will only attempt to forward a notification if a random condition reaches a defined value in order to spreed repetitions of different cars more properly. To ensure that the proposed algorithm works properly we have to examine, how long a vehicle will be within the detection radius of another vehicle and whether this time will be greater or less the sum of waiting times of a vehicle mentioned above.

There are studies in technical literature about possible communication durations between vehicles, e.g. [1]. The scenario supposed by this study presents undisturbed traffic conditions as well as constant velocity of both vehicles, whereby velocity is assumed generally as normally distributed. In the following we present some theoretical considerations extracted from [1] in order to introduce the reader with some important aspects of the classical vehicular traffic theory.

Possible communication durations between vehicles depend on their relative velocity. With higher relative velocity time vehicles have to communicate with each other is getting shorter. Therefore, the worst case scenario is communication between oncoming vehicles. As it will be explained later that kind of communication is necessary for the supposed algorithm in order to avoid dissemination disruptions. It can be shown that relative velocity between vehicles increases when average speed of these vehicles become higher. Thus, the worst case scenario consists of vehicles with a high average velocity of about 130 km/h, e.g. a highway scenario at day. Theoretical examinations start with statistical distribution of velocities. In classical velocity theory values of velocities are generally assumed as normal distributed [2]. Therefore, the probability density function (pdf) of velocity applies to:

$$p_v(v) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \mathrm{e}^{-\frac{(v-\mu)^2}{2\sigma^2}} \tag{3.1}$$

where according to usual notation μ and σ^2 are average value and variance of velocity respectively. Thus, probability distribution function (PDF) results as:

$$P(v \leq V) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \int_0^V \mathrm{e}^{-\frac{(v-\mu)^2}{2\sigma^2}} dv \quad . \tag{3.2}$$

Table 3.1. Typical values of velocity distributions

Scenario	μ [km/h]	σ [km/h]
	30	9
	50	15
day (road)	70	21
	90	27
night (highway)	105	30
day (highway)	130	39
	150	45

Considering two vehicles driving the same direction with v_1 and v_2 as there velocities respectively, where v_1 and v_2 are normal distributed random variables. $P(\Delta v) = P(v_2 - v_1)$ represents the probability of the velocity difference Δv between these vehicles. According to statistical theory Δv is a normal distributed random variable with $\mu_{\Delta v} = \mu_2 - \mu_1$ and $\sigma^2_{\Delta v} = \sigma^2_1 + \sigma^2_2$. The belonging PDF was calculated for $\Delta v > 0$ only.

Figure 3.1 depicts pdf and PDF of Δv using different values of average velocities of the two vehicles and the condition $\mu_2 = \mu_1$. Therefore, pdf is axially symmetric, whereby the highest curves in Fig. 3.1 in both diagrams correspond to the first value in Tab. 3.1. Table 3.2 shows some exemplary

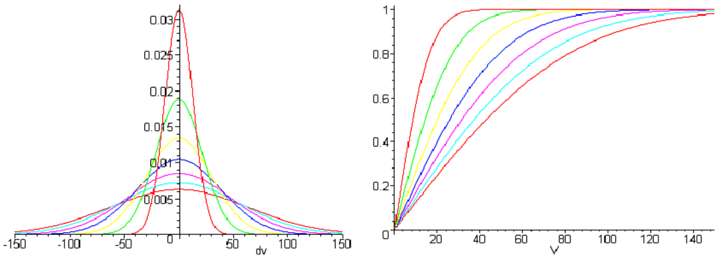


Fig. 3.1. pdf and PDF of velocity differences Δv

values for the PDF of Δv for $V_{\text{avg}} = 70\,\text{km/h}$, $V_{\text{avg}} = 105\,\text{km/h}$ and $V_{\text{avg}} = 130\,\text{km/h}$. These velocities correspond to the day road scenario and the night and day highway scenarios of Table 3.1 respectively. The table confirms the assumption that with higher average velocity the relative velocity between two randomly chosen vehicles is getting higher, too. It can be seen that, e.g. in the road scenario that a probability of relative velocity larger than 50 km/h of vehicles driving the same direction is relatively small with approximately 10 %.

Table 3.2. Exemplary PDF values for Δv

$P(\Delta v \leq V)$			
V [km/h]	$V_{\text{avg}} = 70\,\text{km/h}$	$V_{\text{avg}} = 105\,\text{km/h}$	$V_{\text{avg}} = 130\,\text{km/h}$
10	0,2637	0,1696	0,1439
20	0,4993	0,3317	0,28311
50	0,9077	0,7159	0,6353
80	0,9984	0,9532	0,8564
100	0,9993	0,9679	0,9302

The distance d between two randomly chosen vehicles can be calculated using relative velocity Δv and time t: $d(t) = \Delta v \cdot t$. Thus, d is also a normal distributed random variable. Δv may be positive as well as negative. This can be interpreted as follows:

1. the reference vehicle is overtaking a vehicle ($\Delta v > 0$),
2. the reference vehicle is overtaken by a vehicle ($\Delta v < 0$).

Both cases are identical in practice. Thus, calculations can be limited to $\Delta v > 0$ in case of multiplying pdf and PDF by two. While two vehicles are able to communicate the distance between these vehicles changes from $d = R_{comm}$ to $d = -R_{comm}$, from the reference vehicle's point of view where R_{comm} is the communication range as shown in Fig. 3.2. Thus, the distance passed while communication is possible is $d = 2 \cdot R_{comm}$. Using the condition $\Delta v > 0$ the

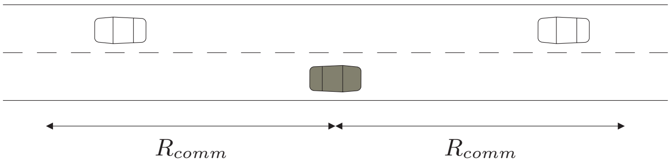


Fig. 3.2. Communication Range R_{comm}

probability distribution function (PDF) of communication duration can now calculated as:

$$p_t(t) = \frac{4 \cdot R_{comm}}{\sigma_{\Delta v}\sqrt{2\pi}} \cdot \frac{1}{t^2} \cdot \mathrm{e}^{-\frac{\left(\frac{2 \cdot R_{comm}}{t} \cdot \mu_{\Delta v}\right)^2}{2 \cdot \sigma_{\Delta v}^2}} \quad \text{for} \quad t \geq 0 \quad . \tag{3.3}$$

Figure 3.3 shows pdf and PDF of possible communication durations Δt_{comm}, $R_{comm} = 1000\,\mathrm{m}$ and with traffic of same direction for different average velocities as presented in Table 3.1. In opposite to Fig. 3.1 this time the first value in Table 3.1 is presented by the lowest curves in both diagrams.

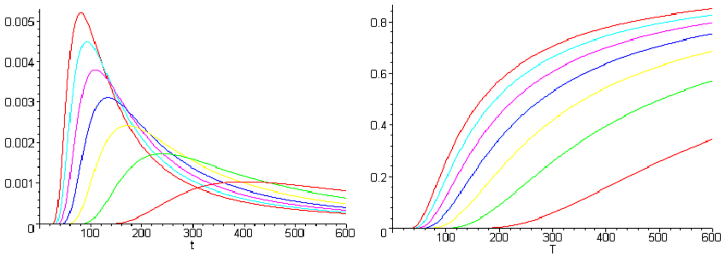


Fig. 3.3. pdf and PDF for Δt_{comm} (same traffic direction)

Considering two vehicles with velocities v_1 and v_2 respectively, where v_1 and v_2 are normal distributed random variables. $P(\Delta v_{opp}) = P(v_2 + v_1)$ represents the probability of the velocity difference Δv_{opp} between these vehicles. According to statistical theory Δv_{opp} is a normal distributed random variable with $\mu_{\Delta v_{opp}} = \mu_2 + \mu_1$ and $\sigma^2_{\Delta v_{opp}} = \sigma_1^2 + \sigma_2^2$. PDF and pdf as represented in Fig. 3.4 correspond to the average speed and variance values of Table 3.1.

To calculate communication durations considering oncoming traffic Δv has to be substituted by Δv_{opp} in (3.3). Figure 3.5 shows pdf and PDF for Δt_{comm}, $R_{comm} = 1000\,\mathrm{m}$ and traffic with opposite directions for different average velocities as presented in Table 3.1 whereas Table 3.3 shows exemplary PDF values for Δv_{opp} and $V_{\mathrm{avg}} = 130\,\mathrm{km/h}$ only, due to it represents the worst case scenario mentioned above.

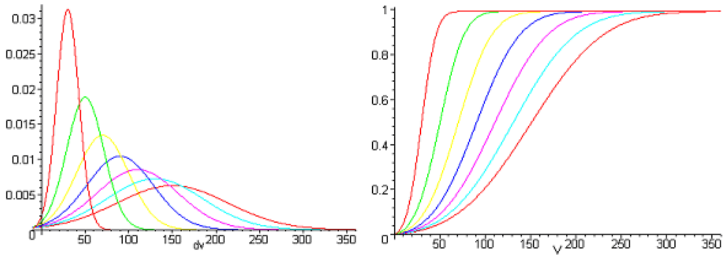


Fig. 3.4. pdf and PDF of velocity differences Δv_{opp}

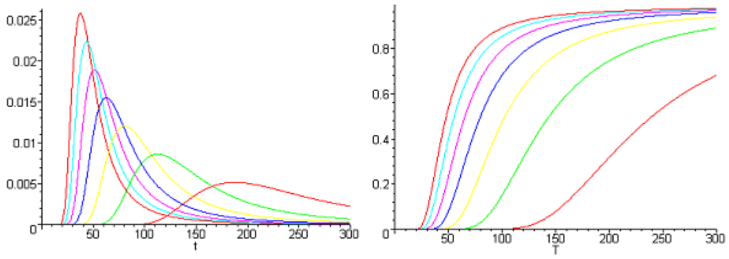


Fig. 3.5. pdf and PDF for Δt_{comm} (oncoming traffic)

Table 3.3. Exemplary PDF values for Δv_{opp}

$P(\Delta v_{opp} \leq V)$	
V [km/h]	$V_{\text{avg}} = 130\,\text{km/h}$
10	0,2637
20	0,4993
50	0,9077
80	0,9984
100	0,9993

If speed increases communication duration decreases as expected. For our purpose it is sufficient to prove that vehicles performing the forwarding algorithm have enough time to communicate with each other in the worst case scenario. Table 3.4 shows some exemplary values of possible communication durations for $R_{comm} = 1000\,\text{m}$ and $V_{\text{avg}} = 130\,\text{km/h}$.

Depending on relative speed it can be seen that communication durations can differ widely. However, despite having oncoming traffic and high average velocity of 130 km/h the probability to have a communication duration less than thirty seconds is about 2 %. That means there is a high probability of communication durations large enough for our goals even in case of the worst case scenario.

Table 3.4. Exemplary values of possible communication durations

T [s]	$P(t \leq T)$	$P(t_{opp} \leq T)$
10	≈ 0	≈ 0
15	≈ 0	$0,5 \cdot 10^{-9}$
30	$0,135263 \cdot 10^{-4}$	0,023054
60	0,029577	0,571938
120	0,276658	0,897809
300	0,663459	0,972690
$>$ 300	0,336541	0,027310

After determining expected communication durations between vehicles we have to estimate the maximum time interval that can be used between periodically transmitted important messages. This can be calculated as follows: We consider a configuration like in Fig. 3.6 with two vehicles, A and B. In succession of its accident vehicle A starts sending Emergency Notifications at time t_{acc} in order to inform other vehicles about the dangerous situation where vehicle B is still out of communication range of A and therefore unable to receive the transmitted message ($t_{acc} < t_{R_{comm}}$). Vehicle B drives toward A and should be warned at least when it reaches the point where the distance between vehicle A and itself is right large enough in order to ensure a reliable reaction of the driver to the accident.

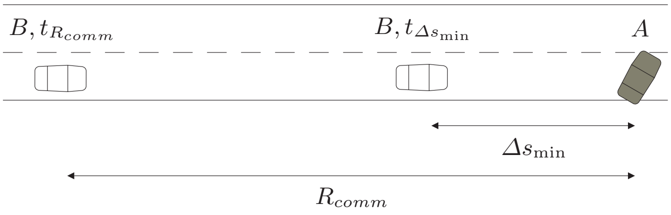


Fig. 3.6. Maximum time interval between periodically transmitted messages

On the assumptions that vehicle B drives with a velocity of $v = 200\,\text{km/h}$ toward the accident point, a total reaction duration of human and machine of $\Delta t_{react} = 1.4\,\text{s}$ and a maximum deceleration of $-a = 5\,\text{m/s}^2$ the minimum braking difference $\Delta s_{\min}$ can be calculated using (3.4) to 387 m. In case of a velocity of $v = 130\,\text{km/h}$ the minimum braking difference decreases to 181 m.

$$\Delta s_{\min} = -\frac{v^2}{2a} + v \cdot \Delta t_{react} \tag{3.4}$$

To calculate the maximum duration between repetitions we have just to consider the difference between time $t_{R_{comm}}$ where vehicle B reaches the communication range of A and time $t_{\Delta s_{\min}}$ where vehicle B reaches the minimum braking difference. Using (3.5) a duration of about $\Delta t_{\max,200} = 11\,\mathrm{s}$ corresponds to $v = 200\,\mathrm{km/h}$ and $\Delta t_{\max,130} = 22.7\,\mathrm{s}$ to $v = 130\,\mathrm{km/h}$ respectively.

$$\Delta t_{\max} = t_{\Delta s_{\min}} - t_{R_{comm}} = \frac{R_{comm} - \Delta s_{\min}}{v} \qquad (3.5)$$

As Table 3.5 shows a maximum deceleration of $-a = 5\,\mathrm{m/s^2}$ is less than we can expect normally. This value as well as the high speed of $v = 200\,\mathrm{km/h}$ was chosen to get an additionally amount of safety.

Table 3.5. Exemplary values of possible decelerations $-a$

	$-a\,[\frac{m}{s^2}]$
ice	1.0
snow	1.5
snow-chain on ice	2.0
bad way	4.0
wet lane	4.5
wet asphalt	5.5
dry lane	6.5
ABS	7.5
very good lane	8.0

After these reflections we want to concentrate on designing the proposed forwarding algorithm. During the first time the success of the algorithm depends on the existence of at least one additionally equipped vehicle within direct communication range of the car damaged in the accident that can start the forwarding process. There are three possible situations we have now to investigate:

1. Because of the accident the transmitter of the vehicle was destroyed before sending at least one Emergency Notification.
2. The vehicle was able to send just one complete message before its transmitter was destroyed.
3. The transmitter is able to keep sending Emergency Notifications periodically because it was not destroyed by the accident.

Item 1 does not need any further investigation because of the algorithm does not apply. Item 3 represents the opposite to Item 1 due to vehicles driving toward the accident point will be warned by the car damaged in the accident itself. Thus, applying the algorithm in order to avoid dangerous situations to other vehicles caused by the accident will be useful but not really necessary. Therefore, special interest has to apply to Item 2, the worst case scenario. We have to consider three different situations (the zone concept will be explained in detail in Sec. 3.2.2):

1. Only vehicles belonging to the Hazardous Zone are involved in the forwarding process.
2. Only vehicles belonging to the Opposite Zone are involved in the forwarding process.
3. Vehicles belonging to both zones are involved in the forwarding process.

We aim to determine whether the algorithm works only with the participation of Hazardous Zone traffic, thus the Emergency Notification is in fact relevant only for these vehicles or otherwise the participation of Opposite Zone vehicles is necessary for the success of the dissemination of Emergency Notifications. For each situation it is supposed that the radio equipment of the crashed vehicle is able to send the Emergency Notification only once so that the worst case scenario applies as described above.

At first we calculate for each situation the probability that at least one equipped vehicle on the studied zone receives the Emergency Notification. It will be made under different circumstances of system penetration rate and traffic densities. Hence, our next objective is to determine the formula which proportionate such probability. Statistical theory describes the probability distribution of net time gaps through the Poisson distribution. The general expression of this distribution shows (3.6):

$$P(X = k) = \frac{\lambda^k}{k!} \cdot \mathrm{e}^{-\lambda} \quad . \tag{3.6}$$

Based on the Poisson Distribution Model for net time gaps between vehicles the probability to find exactly k equipped vehicles in an area of length $L\,[\mathrm{m}]$ with a traffic density of $\rho\,[\mathrm{veh/km/lane}]$ can be calculated using (3.7) and (3.8) where N means the number of lanes while F stands for penetration rate of equipped vehicles:

$$P(X = k) = \frac{n^k}{k!} \cdot \mathrm{e}^{-n} \quad \text{with} \tag{3.7}$$

$$n = L \cdot \rho \cdot N \cdot F \quad . \tag{3.8}$$

For our purposes $L_{\max}$ corresponds to $2 \cdot R_{comm}$ as the reader can already suppose remembering considerations made in previous sections about the distance where communication between two vehicles is possible.

In the particular case of “at least one vehicle” the Poisson Probability Distribution as described above is equivalent to the Exponential Probability Distribution [3, 4]. This supposes a simplification of the mathematical calculation effort. The basic formula that will be used for the evaluation of the algorithm success shows (3.9). It provides the probability to find at least one equipped vehicle in an area L:

$$P(X < L) = 1 - \mathrm{e}^{-(L \cdot \rho \cdot N \cdot F)} \quad . \tag{3.9}$$

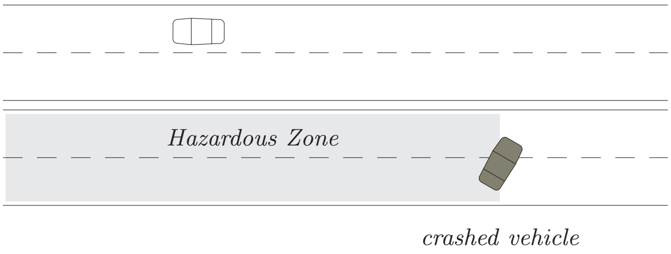


Fig. 3.7. Hazardous Zone

In this case the probability to find at least one vehicle in the Hazardous Zones part of the communication area of the crashed vehicle as shown in Fig. 3.7 that could initiate the spread process of an Emergency Notification is depicted in Fig. 3.8 in dependence of penetration rate. Using $L = R_{comm} = 1000\,\mathrm{m}$, $N = 2$ and (3.9) the calculation corresponds to one driving direction of a highway scenario. As expected the probability increases with higher penetration rate as well as with higher traffic density.

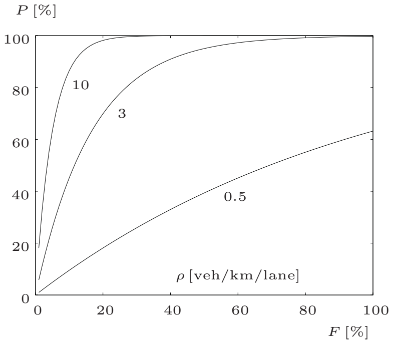


Fig. 3.8. First hop probability using Hazardous Zone traffic only

At the beginning an ad hoc network has to deal with relative low penetration rates. With a penetration rate of, e.g. only 2 % the probability to find at least one vehicle is very small, approximately 2 % in the night scenario with low traffic density which is considered as a worst case scenario for our purposes. The probability increases to almost 12 % in the night scenario with average traffic density and at day time with average traffic density conditions it reaches almost 33 %. The penetration rate should be higher than 40 % in order to obtain the same probability, 33 %, in the worst case scenario (night plus lower traffic density). By average traffic densities at day time the probability is about 86 % with a penetration rate of only 10 %, with 20 % of penetration rate the probability reaches the maximum value. At night time it would be necessary to have a penetration rate higher than 50 % to ensure the reception of an Emergency Notification by a vehicle on the Hazardous Zone.

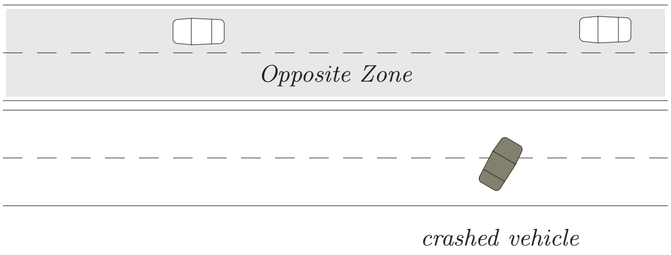


Fig. 3.9. Opposite Zone

After determining the probability to find at least one vehicle in the Hazardous Zone we calculate now the same probability for the Opposite Zone as shown in Fig. 3.9 that could initiate the spread process of an Emergency Notification in dependence of penetration rate. Using $L = 2 \cdot R_{comm} = 2000\,\text{m}$, $N = 2$ and (3.9) the calculation corresponds to the entire opposite driving direction within communication range of the car damaged by the accident in a highway scenario. Figure 3.10 depicts the probability of information propagation over one car within the Opposite Zone for different values of traffic density corresponding to the night and day scenarios respectively. As expected again the probability increases with higher penetration rate as well as with higher traffic density.

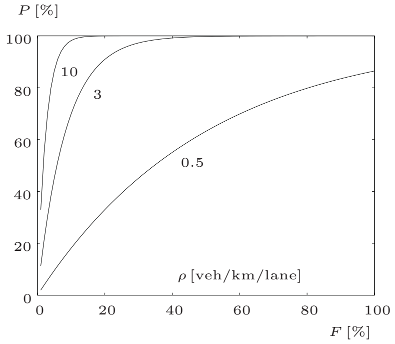


Fig. 3.10. First hop probability using Opposite Zone traffic only

In the night scenario with lower traffic density and a very low penetration rate of only 2 %, the probability to find at least one vehicle is also very small but higher than in the case of a car within the hazardous zone, approximately 4 %. The probability increases to almost 22 % in the night scenario with average traffic density as well as 55 % at day time. The penetration rate should be higher than 40 % in order to obtain the same probability in the worst case scenario (night plus lower traffic density). By average traffic densities at day time the probability is again 40 % with a penetration rate of, this time, only 5 %, with 10 % of penetration rate the probability almost reaches the maximum value. At night time a penetration rate higher than 40 % would be necessary to ensure the reception of an Emergency Notification by a vehicle on the Opposite Zone.

At last we calculate the combined probability for Hazardous and Opposite Zone. Using $L = (1+2) \cdot R_{comm} = 3000\,\mathrm{m}$, $N = 2$ and (3.9) the calculation corresponds to one driving direction of the highway scenario and the entire opposite driving direction within communication range of the car damaged by the accident of the same scenario. Figure 3.11 depicts the probability of information propagation over one car within these zones for different values of traffic density corresponding to the night and day scenarios respectively. As in the cases before the probability increases with higher penetration rate as well as with higher traffic density.

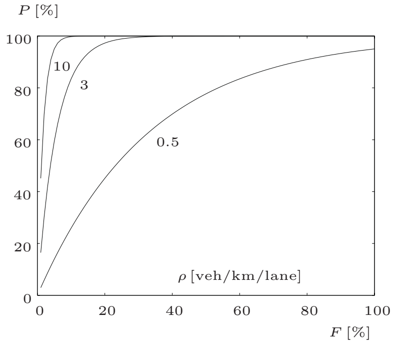


Fig. 3.11. First hop probability using Hazardous and Opposite Zone traffic

In the night scenario with lower traffic density and a very low penetration rate of only 2 %, the probability to find at least one vehicle is still small but higher than in the cases described above, approximately 6 %. The probability increases to 30 % in the night scenario with average traffic density as well as 70 % at day time. The penetration rate should be higher than 40 % in order to obtain the same probability in the worst case scenario (night plus lower traffic density). By average traffic densities at day time the probability is almost 95 % with a penetration rate of only 5 %, with 10 % of penetration rate the probability almost reaches the maximum value. At night time a penetration rate higher than 25 % would be necessary to ensure the reception of an Emergency Notification by a vehicle on the Hazardous or Opposite Zone.

In the following results for the probability of information propagation over multiple hops are depicted with participation of Hazardous Zone traffic only. Two scenarios are considered, a highway scenario with average night traffic conditions ($\rho = 3$ veh/km/lane) and the same highway scenario with average day traffic conditions ($\rho = 10$ veh/km/lane). The probability is calculated for different values of penetration rate. An important conclusion to be extracted from these results is a limit for the penetration rate which can ensure a high probability to propagate an Emergency Notification along a relevant zone of the message with participation of Hazardous Zone traffic only. In fact, only vehicles which belong to this traffic zone are direct affected by an accident and therefore should take the higher responsibility in the propagation process. Participation of vehicles within the other two traffic zones is foreseen in the algorithm in case of adverse dissemination conditions (low traffic density, low penetration rate) in order to avoid disruptions of the communication chain.

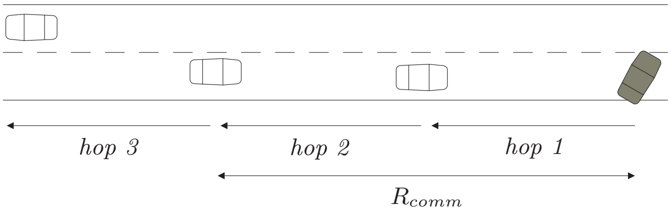


Fig. 3.12. Gain of information range

In the next figures the abscissa axis illustrates the gain of information range. We have studied a simplified scenario as depicted in Fig. 3.12 where each new hop of a message represents a gain of information range of 500 m, i.e., half of maximum transmission range. Gain "0" corresponds with the first hop probability already shown in Fig. 3.8. For each hop it is assumed that the receiving vehicle places 500 m away from the transmitting vehicle, this value is taken as the smallest distance which ensures a relative high usage of a messages forwarding as shown in Fig. 3.20.

Figure 3.13 shows results obtained for the night scenario while Fig. 3.14 shows results for the highway scenario at day time.

In the night scenario with average traffic densities the penetration rate that ensures a high propagation of information is about 50 % using Hazardous Zone traffic only. With this penetration rate the algorithm ensures a dissemination of the message over 7 km, which corresponds to 14 hops with a probability of 50 % and over 3.5 km (7 hops) with a probability of 70 %.

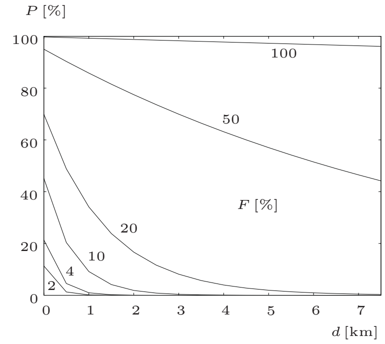


Fig. 3.13. Probability of information propagation over multiple hops (night)

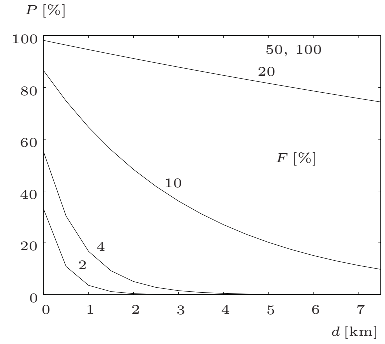


Fig. 3.14. Probability of information propagation over multiple hops (day)

In the day scenario with average traffic conditions the algorithm deals with a higher traffic density. Thus, the penetration rate limit which already ensures a high probability of information propagation is only 20 %. With this penetration rate the algorithm ensures a dissemination of a message over 7 km, which corresponds to 14 hops, with a probability of almost 80 %.

3.2.2 Traffic Zones

After discussing the question under which circumstances communication among vehicles will be successful regarding time in Sec. 3.2.1 the following section will now deal with special qualities relating to space conditions.

The algorithm proposed in this work allows us to enlarge the area within the Zone of Relevance of the accident in which a vehicle could receive an Emergency Notification. A definition of the Zone of Relevance concept shows Fig. 3.15. In this application vehicles are provided with a radio equipment allowing them to contact with other equipped vehicles in their surrounding area. No fixed infrastructure to support the communication is assumed and the resulting ad hoc network requires no additional infrastructure at the road side. The vehicles use omni directional antennas implying that a sender can transmit to multiple hosts simultaneously. Many vehicles do or will soon utilize navigation systems like the Global Positioning System (GPS). Thus, it is assumed that equipped vehicles know their location more or less accurately. Furthermore, to make the algorithm work, vehicles need to be aware of their current locations. Taking the driving direction into account, a vehicle can distinguish more reliably whether it is approaching a special point or not as well as if it employs a digital road map it may improve its ability to classify a given situation.

In case of an accident vehicles driving toward or already into the hazardous area should be warned by the crashed vehicle. The goal is to disseminate information about the accident quickly and efficiently to any vehicle affected by the dangerous situation.

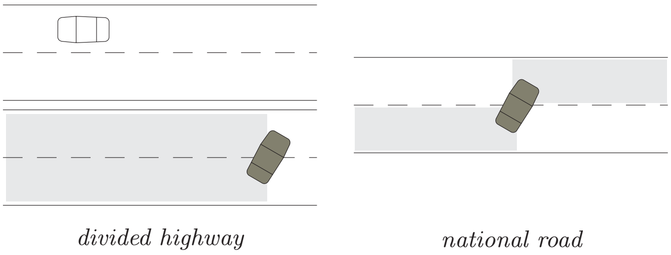


Fig. 3.15. Relevance Zone and road types

Two different road types are considered: A divided highway and a national road that can be seen as a highway without of any kind of physical divider and with fewer number of lanes. For the road model of a divided highway the Zone of Relevance covers the region behind the accident on the side of the highway where the accidents happens. On divided highways an accident usually does not harm vehicles of the other driving direction. In case of the second road type, the national road, vehicles having an accident can affect both driving directions. Hence, all vehicles approaching the position of the accident are part of the Relevance Zone. For our forwarding strategy we suppose to divide a road into three different zones related to the point of accident:

1. Vehicles belonging to the Hazardous Zone drive toward the point of accident. They can be involved in the accident directly because there is no physical divider between these cars and the accident.
2. Vehicles belonging to the Opposite Zone drive away from the point of accident in general. If there is a physical divider on the road the Opposite

Zone covers the entire opposite driving direction. Thus, in this case there are vehicles that drive toward the point of accident as well. Vehicles within this zone can play a very important role in the dissemination process, helping to avoid disruptions of the communication chain in case of long net time gaps between equipped vehicles.

3. Vehicles belonging to the Neutral Zone drive always away from the point of accident. The concept Neutral Zone is used because vehicles within this area are not affected by the accident directly. However, their participation in the dissemination mechanisms may be necessary under special traffic conditions, e.g., a very low traffic density or system penetration rate.

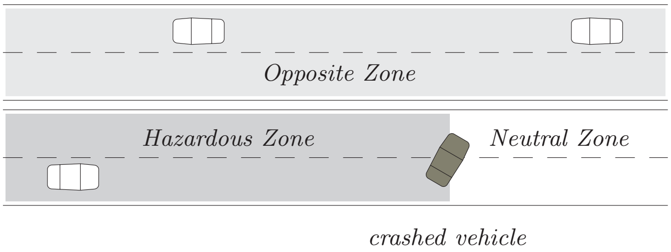


Fig. 3.16. Traffic Zones in a highway scenario with physical divider

Figure 3.16 shows the classification of road lanes and traffic into different zones of a divided highway. If there is no divider like in Fig. 3.17 the Neutral Zone disappears due to the inability of assigning a vehicle that caused an accident to a certain driving direction.

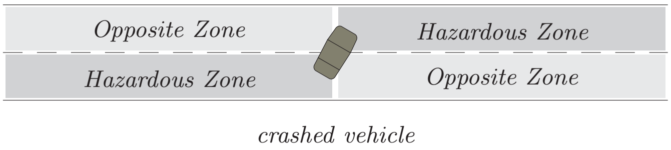


Fig. 3.17. Traffic Zones in a national road scenario without physical divider

As it can be seen by comparing Fig. 3.15 with Fig. 3.16 and Fig. 3.17 respectively Relevance Zone and Hazardous Zone cover the same area because vehicles in theses zones are always affected by an accident directly. Both terms represent different concepts. While Hazardous Zone stands for the traffic's point of view Relevance Zone means the communication's point of view.

3.2.3 Rules and States – the Forwarding Algorithm

Before discussing the proposed forwarding algorithm, the main goal of this section, we want to discuss the concept "Transmission Range" vs. "Information Range" shortly because of their relevance for the algorithm.

Predominantly the Transmission Range of a vehicle depends on its antenna's transmission power, antenna's height and on the propagation channel characteristics. The size of an area covered by a radio system directly depends on these factors. As already mentioned before it is assumed vehicles use omni directional antennas, i.e., their radio coverage areas extend homogeneously in a circle with the respective transmitting vehicle as center and its transmission range as radius. Under normal circumstances, a vehicle sending one data packet can reach all vehicles within its transmission area simultaneously. Information Range is a definition related to one message and tells us how far from the source of a message has arrived the information of this message at all. When using a multi-hopping strategy the Information Range of a message should be larger than the Transmission Range of the initiating vehicle as shown in Fig. 3.18. However, under conditions of a very high interference level vehicles may receive a signal but it could be impossible for them to decode the message. In this case Information Range is smaller than Transmission Range.

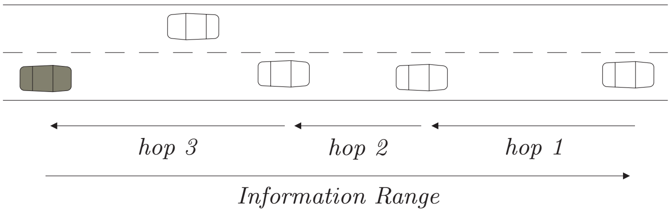


Fig. 3.18. Information Range

The proposed algorithm aims to extend the Information Range of a message under normal circumstances of a channel interference level. That is achieved by multi-hopping a message among equipped vehicles. First, a strategy has to be defined in order to prevent a message from being forwarded infinitely. Many protocols define a maximum number of hops so that if the number of hops made by a message or packet exceeds a given threshold the system discards the packet. In this work this strategy is not used, but another based on the importance of a message for the recipient and the usage of a message repetition for the dissemination process. The importance of a message depends on its content as well as on the distance to its place of origin. Therefore, a function can be defined in order to describe the importance i of a message in dependence on a distance d, e.g. $i(d_{acc})$ in case of an accident for an Emergency Notification, as shown in Fig. 3.19. D_{RZ} means the distance between accident and the end of the Relevance Zone whereas $\Delta s_{\min}$ stands for the minimum braking difference as described in Sec. 3.2.1. A vehicle discards a received message only when its importance reaches the value “0”.

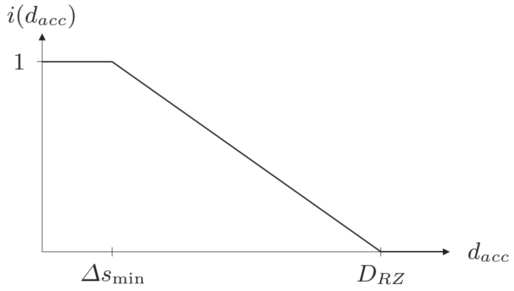


Fig. 3.19. Importance of a message

Like in case of the importance of a message a function can be defined that describes the usage u of a message repetition in dependence on the distance between a receiving vehicle $n+1$ and a transmitting vehicle n. As Fig. 3.20 shows $u(d_{n,n+1})$ depends not on the place of origin of the message but only on the distance $d_{n,n+1}$ between the vehicles. R_{comm} stands for communication range as shown in Fig. 3.2.

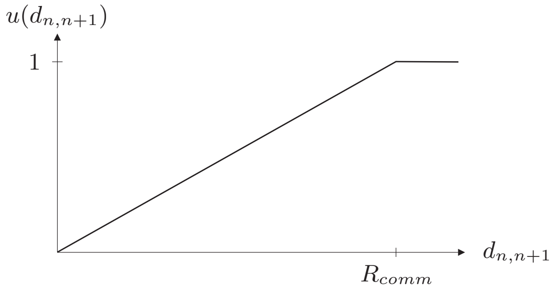


Fig. 3.20. Usage of a message repetition

The algorithm distinguishes between two kinds of messages:

1. An “Emergency Notification” initiated by a vehicle damaged in an accident. As already discussed in Sec. 3.2.1 it is assumed that the vehicle was able to send at least one complete message.
2. A “Forwarded Message” sent by a vehicle involved in the dissemination process of the Emergency Notification.

In a first step the algorithm determines whether the received message is currently unknown for the vehicle or not. Due to multi-hopping it is more likely to receive transmissions of the same message which have to be discarded. In order to determine whether a message is already known each vehicle implements a list of recently received messages. This list stores only one copy of each received message. Already known messages just remain in the list until its importance reaches the value “0” while unknown messages are added to the list. An Emergency Notification with an internal counter $I_{count} = 0$ was always sent by the vehicle damaged in the accident due to it is the original message whereas an Emergency Notification with $I_{count} \geq 1$ is in fact a Forwarded Message that was sent by a vehicle involved in the forwarding process. An Emergency Notification with $I_{count} = 0$ may be received more than once. In this case the radio equipment of the vehicle that caused the accident was not destroyed and therefore still capable of keeping sending the message periodically.

When a vehicle receives a new Emergency Notification the forwarding algorithm determines the probability of resending the received message. The probability of forwarding a received message depends on a list of factors:

1. To which driving zone the vehicle belongs. The more dangerous its position is the higher is the probability of forwarding a message. Thus, vehicles within the Hazardous Zone have a higher forwarding probability than vehicles within any other zones. The Opposite Zone has the next higher probability for its vehicles whereas the lowest forwarding probability corresponds to vehicles of the Neutral Zone.
2. The distance d_{acc} between vehicle and accident point.
3. The distance $d_{n,n+1}$ between receiving vehicle $n + 1$ and forwarding vehicle n in case of a Forwarded Message.

If the vehicle decides not to resend the Emergency Notification it puts the message on its internal stack. Thus, it is still able to forward the message later. The vehicle reaches a “Wait for Forward” state where it tries to resend a message periodically as long as no confirmation messages from other vehicles are received. When a Forwarded Message about the same event that the vehicle tries to propagate is received the received message is considered as an acknowledgment. The aim of this behavior is to give the algorithm more resistance against disruptions of the communication chain as well as to avoid unnecessary overflooding of the underlying radio network. In case of not receiving any other Forwarded Message about the Emergency Notification, which could happen if only this vehicle was within transmission range of the crashed vehicle, it is still guaranteed that the vehicle tries to resend the Emergency Notification until the relevance of the message reaches the value “0”.

State 1. Wait for Forward
Reached when an Emergency Notification was received. A decision whether to forward or not will be made after 3 s:

In case of “yes”:

- if the Emergency Notification was not forwarded send message and go to “Wait for Acknowledge” state, if “channel not free” try again after 1 s;
- if the Emergency Notification was already forwarded make a new decision after 6 s.

In case of “no”:

- if the Emergency Notification was not forwarded make a new decision after 3 s;
- if the Emergency Notification was already forwarded make a new decision after 6 s.

In case of “cancel”:

- discard message and delete it from stack.

If the vehicle decides to forward the message it reaches a “Wait for Acknowledge” state. Like in “Wait for Forward” state as long as no confirmation messages from other vehicles are received, that ensures the vehicle that the dissemination process goes on, it will initiate to resend again the message periodically. In general the number of periodical transmissions is unlimited. The vehicle stops with the periodically forwarding of the message when importance becomes “0”. Again, the aim of this behavior is to give the algorithm more resistance against disruptions of the communication chain. The last informed vehicle does not leave its forwarding task until the relevance of the message reaches the value “0” or it is sure another vehicle goes on with the dissemination process.

State 2. Wait for Acknowledge
Reached when a received Emergency Notification was forwarded. A decision whether to repeat or not will be made after 6 s:

In case of “repeat”:

- send message again, if “channel not free” try again after 1 s and
- remain in “Wait for Acknowledge” state, make a new decision after 6 s.

In case of “acknowledged”:

- discard message and delete it from stack.

As described in Sec. 3.2.1 the maximum duration between periodical repetitions of Emergency Notifications in case of $v = 200\,\mathrm{km/h}$ corresponds to $\Delta t_{\max,200} = 11\,\mathrm{s}$. We have chosen a maximum duration value of $\Delta t_{\max} = 6\,\mathrm{s}$ in order to get an additional amount of safety.

After finishing the forwarding task the behavior of the vehicle is identical as before. In case of forwarding a message the vehicle enters the Wait for Acknowledge state again and in case of not forwarding the vehicle enters the Wait for Forward state. Receiving an acknowledge from other vehicles about the message the vehicle tries to disseminate means in both cases that the forwarding task of the vehicle is for the moment no more necessary for the global performance of the algorithm.

Controlled by multiple dependencies of the forwarding probability the algorithm determines that not each vehicle that receives a message resends it automatically but only those whose relevance in the dissemination process is the highest. Other vehicles wait and observe the progression of the spread process. Thanks to the existence of more than one resending attempt within the forwarding algorithm and in case of their participation become necessary, they get a chance of forwarding the message by themselves periodically. This performance aims again to avoid overflooding of the underlying radio network with unnecessary messages.

Messages containing safety relevant information are most important to vehicles belonging to the Hazardous Zone. Thus, forwarding of such messages should be done by these vehicles first. However, as described in Sec. 3.2.1 a participation of vehicles belonging to other zones still remains necessary at least in cases of low penetration rate or low traffic density.

3.2.4 Traffic Models

Inter-vehicle communication represents a distinctive case for ad hoc networks, characterized by high speed and "one-dimensionality" of their scenarios. By modeling of vehicles movements it can be assumed for small traffic densities that cars are moving independently of each other. At high traffic densities the complex interactions among neighboring vehicles make modeling of such a dynamical system a challenge.

For low density scenarios basic traffic models are usually built by specifying probability distributions for vehicle speeds and net time gaps. These net time gaps provide a safety-related measurement of distances and are typically measured in seconds. However, for our purposes, it is more interesting to know the distance between vehicles in meters. In this way, the probability for a vehicle to reach at least one other equipped vehicle within a distance d was modeled by an exponential distribution [5]:

$$P(x < d) = 1 - \mathrm{e}^{-\frac{d}{d_a}} \quad , \tag{3.10}$$

where d_a is the average effective distance between two equipped vehicles. The value of d_a depends on the average velocity V_{avg}, the net time gap Δt, the number of lanes N and the penetration rate F:

$$d_a = \frac{V_{\mathrm{avg}} \cdot \Delta t}{N \cdot F} \quad . \tag{3.11}$$

For a more accurate modeling, in particular for high density scenarios, some microscopic traffic simulation models have been proposed and analysed for their appropriateness within the framework, e.g. Psycho-physical models [5], Car-following models [6], Velocity-density models [7], Cellular Automaton Based models [8] or Drive-based models [9]. These models, in their basic versions, model the speed or acceleration with what a vehicle must perform in order to keep constant the distance between it and the leading vehicle.

The cellular automaton approach was selected for simulations introduced in Sec. 3.3, because it provides sufficient accuracy for low computational costs. For the sake of simplicity, we do not model complex maneuvers like lane changes or overtaking. Cellular automata are discrete models that are consist of an infinite, regular grid of cells, each in one of a finite number of states. The grid can be in any finite number of dimensions. Time is also discrete, and the state of a cell at time $t + 1$ is a function of the state of a finite number of cells at time t. In order to describe a road using a cellular automaton cells are defined as 7.5 m long. This corresponds to the space required by a vehicle in traffic jam. Each cell may be empty or engaged by exactly one vehicle. Vehicles are characterized by their current velocity v. The velocity can be one of the allowed integer values $v = 0, 1, 2, \ldots, v_{\max}$. In a simple case $v_{\max}$ corresponds to a speed limit and is therefore equal for all vehicles. If a vehicle is present in the cell it may be advanced to another cell using a simple rule set. A typical configuration of a cellular automaton is shown by Fig. 3.21.

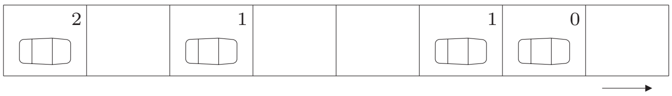


Fig. 3.21. Configuration at time t

The state of the road at time $t+1$ can now be obtained from that at time t by applying the following rules to all cars at the same time:

Step 1. Acceleration
If speed v_n of vehicle n is less than the maximum speed $v_{\max}$, increase vehicle's speed by one cell per time step: $v_n \rightarrow \min\{v_n + 1, v_{\max}\}$.

Step 2. Braking
If speed v_n of vehicle n is greater than the number of empty cells d_n in front of it, set vehicle's speed to the number of empty cells: $v_n \rightarrow \min\{v_n, d_n\}$.

Step 3. Randomization
If speed v_n of vehicle n is greater than zero, decrease vehicle's speed by one cell per time step with a probability of p: $v_n = f(p) \rightarrow \max\{v_n - 1, 0\}$.

Step 4. Driving
Move vehicle n forward the number of cells given by vehicle's speed v_n: $x_n \rightarrow x_n + v_n$.

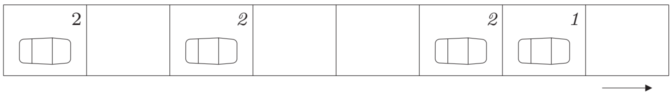


Fig. 3.22. Acceleration

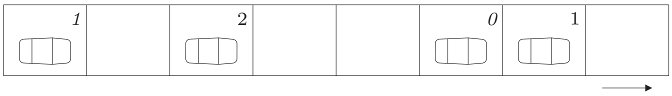


Fig. 3.23. Braking

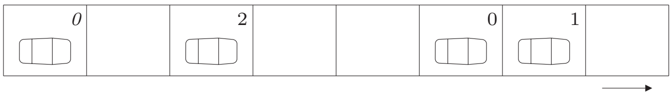


Fig. 3.24. Randomization

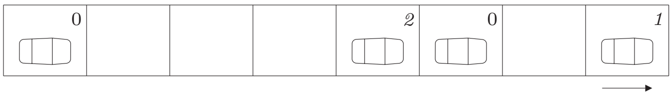


Fig. 3.25. Driving = configuration at time $t + 1$

The applying of a minimal rule set is shown by Fig. 3.21 to Fig. 3.25. Minimal rule set means there is no negligible rule within the set. Thus, a realistic behavior with fewer rules is not possible where realistic behavior stands for spontaneous appearance of traffic jam and the right form of the fundamental diagram, i.e. the correlation between traffic density and traffic flow. Even the change of the order in the update procedure leads to a completely different behavior. In case of low traffic densities traffic flow is proportional to traffic density because of the lack of interaction among the vehicles. In case of increasing traffic densities interaction among vehicles becomes more important. Thus, the characteristic of the correlation between density and flow is getting more and more non linear. Finally, interaction among vehicles becomes dominant so that traffic flow decreases while traffic density still increases.

3.3 Car-to-Car Forwarding: Performance Analysis

After introducing the reader with the basic concepts related to this work in the sections before we want to show now some results obtained by our simulations.

3.3.1 Results by Using Different Zone Concepts

In the following we present simulation results obtained by using the parameter presented in Table 3.6. Vehicles of all three zones are involved in the forwarding process. The term "forwarding" stands for the number of first repetitions of a message by the vehicles whereas "repeating" means that vehicles had to repeat transmissions more than once due to the lack of an acknowledgment. A third value "max" shows the maximum number of the internal message counter I_{count} as explained in Sec. 3.2.3, i.e. "max" stands for the highest length of a repetition chain and therefore for the maximum number of hops of a message that occurred in a simulation.

Table 3.6. Parameter for long time simulations

Parameter	Value
Number of vehicles	1200 (600/lane)
Simulation duration	3600 s
Number of lanes	2
Road length	10 km (1333 cells)
Length of Relevance Zone $D_{RZ,HZ}$	4.5 km (600 cells)
Length of Relevance Zone $D_{RZ,NZ}$	1.125 km (150 cells)
Communication range R_{comm}	1 km (133 cells)
Guaranteed receiving of EN	0.5 km (67 cells)
Initial speed	108 km/h
Maximum speed $v_{\max}$	162 km/h
Penetration rate F	variable
Number of sent EN	1 (worst case scenario)
Cell of accident	99
Time of accident t_{acc}	20 s
Time to wait until forwarding	3 s
Time to wait until repeating	6 s
Time to wait if "channel not free"	1 s

Although the simulations introduced below cover a range from 1 veh/km/lane to 30 veh/km/lane there are three values of traffic flow indicated by special scenarios as described in Sec. 3.2.1 that we want to spend a little bit more attention:

1. The "Night Scenario" is defined by a traffic flow of about 3 veh/km/lane. In this scenario the probability to reach another vehicle with just one

sent message is very low due to the low probability of having at least one equipped vehicle within communication range of the initiating vehicle at sending time t_{acc}.

2. The "Day Scenario" is defined by a traffic flow of about 10 veh/km/lane. In this scenario the probability to reach at least one other vehicle with just one sent message should be reasonably high.
3. Like the other scenarios the "High Traffic Flow Scenario" is defined by a traffic flow of about 18 veh/km/lane. In this scenario the probability to reach at least one other vehicle with just one sent message should be almost "1".

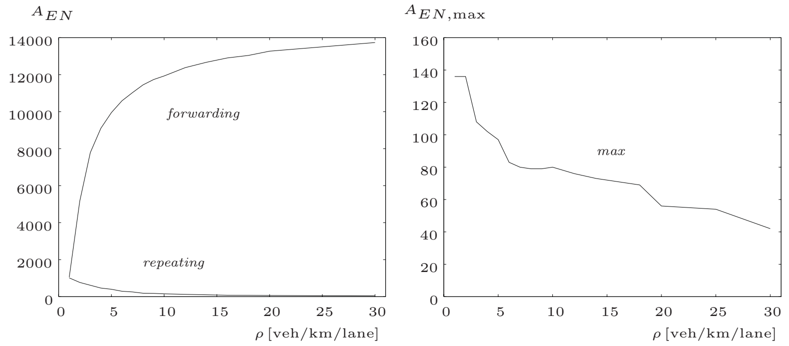


Fig. 3.26. 100 % Penetration rate

As it can be seen in Fig. 3.26 there are no problems that could be expected if the penetration rate equals to 100 %. There are a large number of "forwarding" repetitions of different vehicles even in case of low traffic density, e.g. 1052 repetitions and a traffic density of 1 veh/km/lane. The number of multiple transmitted messages, as explained above they are marked by the term "repeating", are about in the same size. If traffic flow increases the number of "forwarding" messages increases as well while the number of "repeating" messages decreases very fast due to the fact that with higher traffic densities it is no longer necessary by a vehicle to transmit a Forwarded Message more than once in order to get an acknowledgement by another vehicle (proportion forwarding : repeating equals to 13572 : 45 in case of 30 veh/km/lane). Despite using a conservative set of rules for message repetition as introduced in Sec. 3.2.3, there is still a large number of repeated messages in case of high traffic flow, e.g. 40 messages/min/km. Thus, rule settings have to be a focus on further investigations to reduce this large number of repetitions.

Figure 3.26 right shows the number of maximum hops made by a message. As mentioned above in case of forwarding a message the internal counter of this message is incremented by the forwarding vehicle before transmitting:

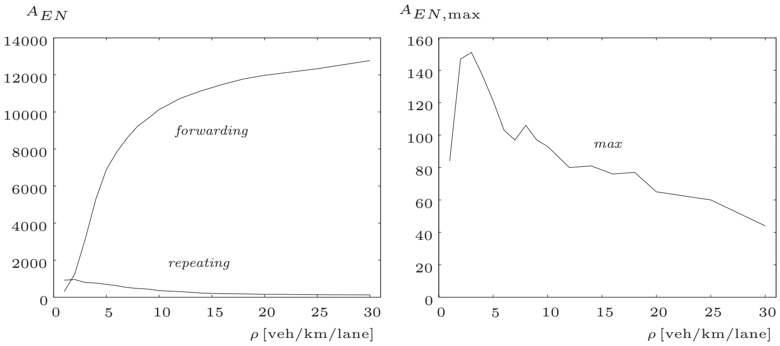


Fig. 3.27. 50 % Penetration rate

$I_{count} \rightarrow I_{count} + 1$. A counter value of "0" indicates an original message sent by, e.g. a car involved in an accident whereas a value of "1" identifies a first repetition of this message by a vehicle that received the original message. A third vehicle that receives this first repetition increases I_{count} to "2" and so on. Thus, Fig. 3.26 right shows $I_{count,\max}$ obtained by given traffic flows respectively. It can be seen that the length of repetition chains decreases with increasing traffic flow and therefore reaches its maximum under low traffic conditions.

There are no significant differences between 100 % and 50 % penetration rate except for low traffic. For the first time vehicles require multiple repetitions of a message in order to accomplish their forwarding tasks.

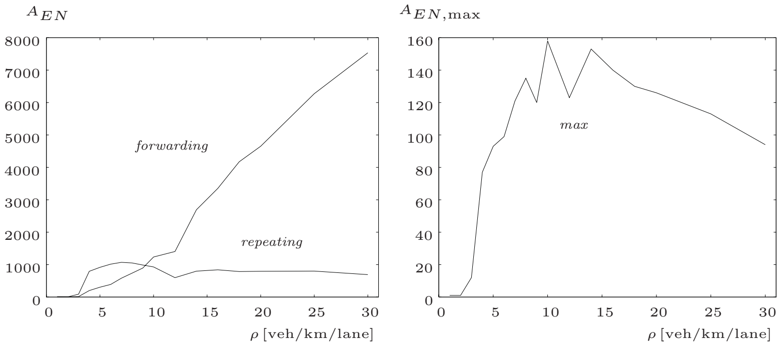


Fig. 3.28. 10 % Penetration rate

Figure 3.27 shows that the maximum value of "max" shifts to higher traffic flows. As it can be seen below this behavior continues when penetration rate decreasing goes on.

At a rate of 10 % we are first within a realistic range of penetration rate in the foreseeable future. It can be seen that vehicles require multiple repetitions of messages in order to accomplish their forwarding tasks. Until to a traffic flow of about 9 veh/km/lane the part of multiple repetitions outbalanced the part of single repetitions clearly.

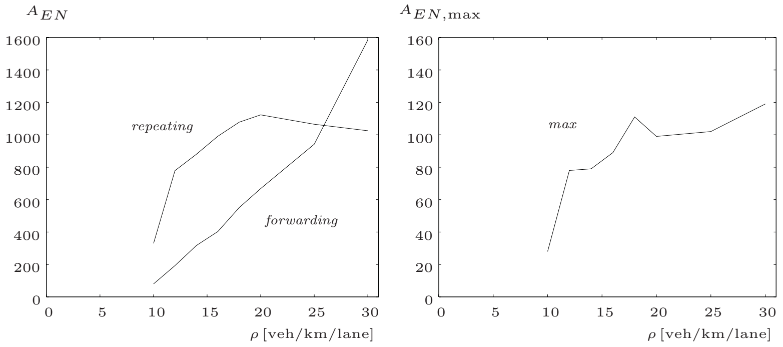


Fig. 3.29. 4 % Penetration rate

As the graph in Fig. 3.28 shows there is an optimum regarding maximum length of repetition chain in the range of 10 to 12 vehicles per km and lane.

In the range of less than 10 % penetration rate, e.g. as shown in Fig. 3.29 and Fig. 3.30 for 4 % and 2 % respectively, vehicles are no longer able to proceed their forwarding tasks until penetration rates reach values like in the day scenario.

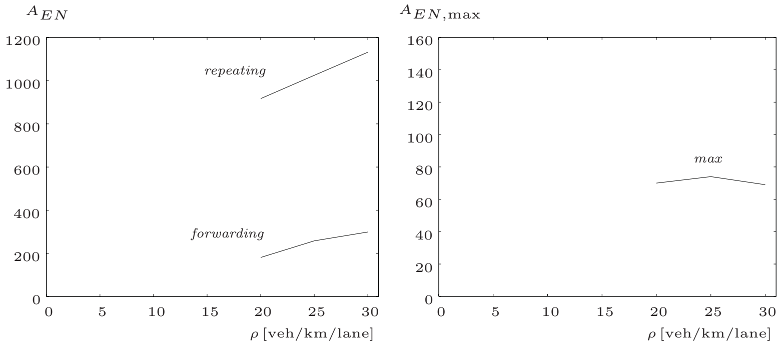


Fig. 3.30. 2 % Penetration rate

As mentioned above in case of only 2 % penetration rate vehicles are no longer able find an appropriate successor in the forwarding process until traffic flow increases dramatically. Even in these cases multiple repetitions occurring much more than single repetitions.

3.3.2 Message Distribution

In the following we want to consider the distribution of Forwarded Messages sent by a vehicle in dependence on the position of this vehicle. The position corresponds to a cell number as described in Sec. 3.2.4. In the diagrams above points are used to identify the number of messages whereas a line is used for belonging average values calculated using a sliding window of 10 cells, for some example combinations of penetration rate and traffic flow. The initiating Emergency Notification is always transmitted at cell 99. Due to vehicles drive in a circle, cell 0 is successor of cell 1333. As shown in Sec. 3.3.1 used penetration rates are unrealistic high. However, these rates are chosen in order to describe the behavior of vehicles in the forwarding process more clearly.

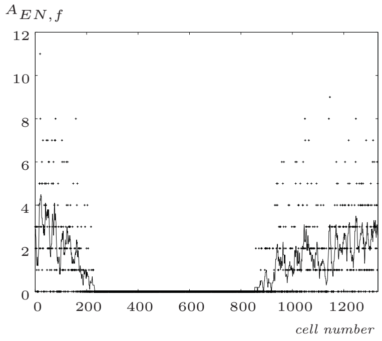


Fig. 3.31. 10 % Penetration rate, 10 veh/km/lane

As it can be seen in Fig. 3.31 and Fig. 3.32 respectively the behavior of vehicles that repeating a received message is already visible. Starting at cell 800 the number of forwarded messages increases continuously.

The maximum value of repeated messages is located in front of the accident as expected in order to inform vehicles that driving toward this position right on time.

Increases the number of single repetitions, e.g. in case of a higher level of equipped vehicles as shown in Fig. 3.33, vehicles involved in the forwarding process changes their behavior apparently.

While in case of 100 % penetration rate and a traffic flow of 4 veh/km/lane transmitted messages are distributed continuously over all cells within the Relevance Zone this proportion changes when message density increases as

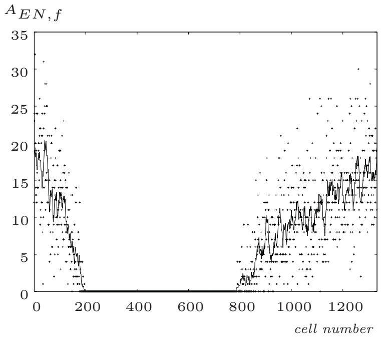


Fig. 3.32. 10 % Penetration rate, 30 veh/km/lane

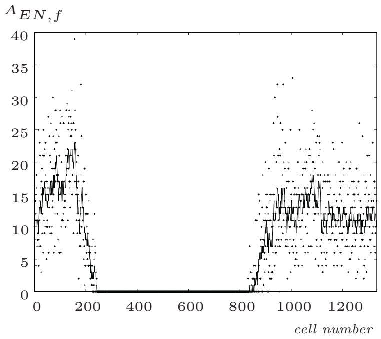


Fig. 3.33. 100 % Penetration rate, 4 veh/km/lane

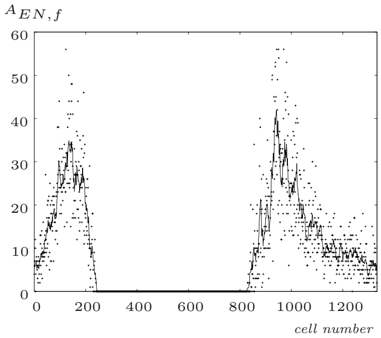


Fig. 3.34. 100 % Penetration rate, 10 veh/km/lane

shown in Fig. 3.34 and Fig. 3.35. In these cases the maximum value of forwarded messages shifts toward the end of the Relevance Zone.

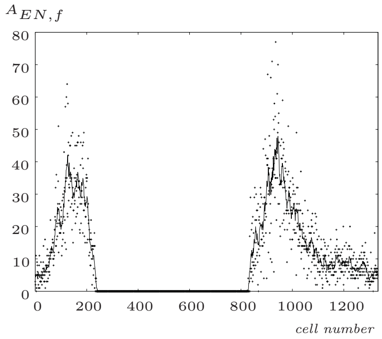


Fig. 3.35. 100 % Penetration rate, 18 veh/km/lane

As it can be seen in the figures above there is still a demand of optimizing the behavior of vehicles involved in the forwarding process regarding the large number of transmitted messages as well as the distribution of messages over all positions.

3.3.3 First Repetition Probability

Having discussed the number of forwarded messages and their distribution we want now simulate the probability to find at least one equipped vehicle within the communication range of a randomly chosen vehicle. This probability was already calculated in Sec. 3.2.1 using a simple approach. In case of the worst case scenario described in the same section this question is very important due to it corresponds to the decision whether the forwarding process may start or not at all. In contrast to this fact vehicles following in the repetition chain can repeat a received message as often as necessary because of, e.g. they are not affected by an accident like an initiating vehicle may be. Thus, the probability to find at least one equipped vehicle does not matter for these vehicles. The simulation results introduced below are obtained by using the parameters presented in Table 3.7.

Figure 3.36 shows results where vehicles of the Hazardous Zone are involved in the forwarding process only. It can be seen that for low penetration rate receiving of the original message even in case of a high traffic density of 10 veh/km/lane is still insufficient. On the other hand using, at least for a start up phase, unrealistic high values of penetration rate but a low traffic flow value of 3 veh/km/lane results in a probability of about 70 % for a successful first forwarding task. The results obtained in the short time simulations

Table 3.7. Parameter for short time simulations

Parameter	Value
Number of vehicles	1200 (600/lane)
Simulation duration	120 s
Number of lanes	2
Road length	10 km (1333 cells)
Communication range R_{comm}	1 km (133 cells)
Guaranteed receiving of EN	0.5 km (67 cells)
Initial speed	108 km/h
Maximum speed $v_{\max}$	162 km/h
Penetration rate F	variable
Number of sent EN	1 (worst case scenario)
Cell of accident	variable
Time of accident t_{acc}	90 s

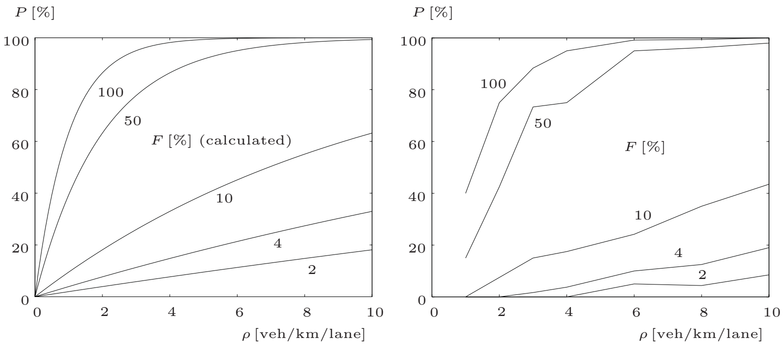


Fig. 3.36. Forwarding by vehicles of Hazardous Zone only

approve the assumption as discussed in Sec. 3.2.1 that taking the forwarding process by vehicles of the Hazardous Zone only leads to disadvantageous values of forwarding probability.

Results obtained by using all available equipped vehicles to take part of the forwarding process are shown in Fig. 3.37. In comparison to the scenario of using vehicles of the Hazardous Zone only there can be observed significant improvements for all penetration rates. In case of traffic flow values greater or equal 50 % the probability of a successful receiving of the original message corresponds to almost 100 % even in case of a low traffic flow of 1 veh/km/lane. Furthermore, having realistic penetration rates, except in case of 2 %, and a traffic flow of at least 6 veh/km/lane sufficient values for receiving of an original message are already very likely.

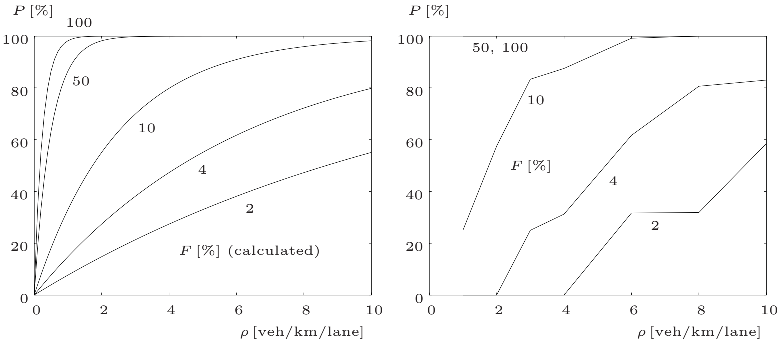


Fig. 3.37. Forwarding by vehicles of all three zones

3.3.4 Fastest Direct Sequence Forwarding and Information Coverage

Having discussed the question of forwarding messages in general as well as the probability to find at least one equipped vehicle within the communication range of a randomly chosen vehicle in the sections before it can be considered additionally the question how fast a message may be transfered from the place of its origin to the end of the Relevance Zone. As a required condition in this case all messages involved in the task of information transport have to belong to the same repetition chain. This "Fastest Direct Sequence Forwarding" is important for extending the underlying forwarding algorithm from 1- to n-dimensionality.

In opposite to the first part of this section we want to consider in the second part the question how long information contained in a certain message may remain in a geographical area like a Relevance Zone without using of any kind of fixed infrastructure at the road side. What the term "Information Coverage" of a certain zone stands for is depicted in Fig. 3.38. Two vehicles are shown that repeat a received message at the same time independently of each other. Thus, they provide an area with information much larger than the area covered by the communication range of just one vehicle. Due to the almost periodically appearance of such message repetitions in similar configurations information about events can remain in their respective target areas over time even when vehicles which carried the information before leave these areas.

As it could be seen in different simulations a successful forwarding of messages using Fastest Direct Sequence Forwarding only is possible but not very likely. Providing of an area with information about an event is the much easier task even in cases of disadvantageous conditions.

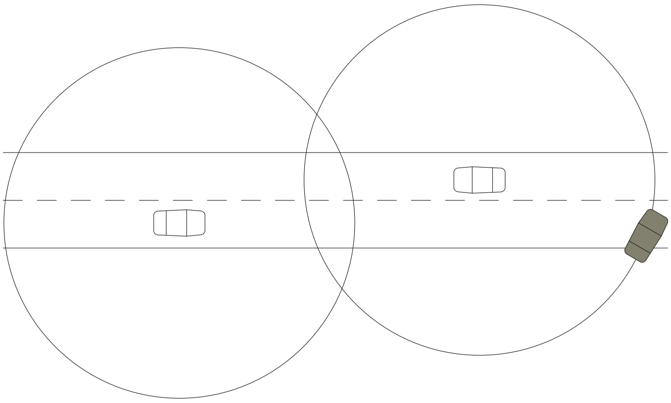


Fig. 3.38. Information Coverage

3.4 Conclusions

In this chapter we introduced an algorithm for the propagation of Emergency Notifications, i.e., messages released automatically by vehicles in emergency situations, like airbag ignition or hard braking. The algorithm is based on a multi-hopping strategy. In this work, the area of interest for a message dissemination was called Relevance Zone of a message. The algorithm is based on the division of different traffic zones, message importance, and usage of message repetition. In ad-hoc networks resource management techniques are very important due to the limitation of resources. Thus, the proposed algorithm aims to disseminate an Emergency Notification quickly among vehicles of the Relevance Zone as well as minimizing the hazard of congesting the underlying radio network.

The forwarding of a message was based on multiple criteria to ensure that vehicles forward messages only in case of high benefits for the dissemination process. Vehicles within the Hazardous Zone take the highest responsibility in the dissemination process, due to their belonging to the Relevance Zone. Vehicles within other traffic zones help to avoid disruptions of the propagation chain under adverse circumstances of low traffic density or low penetration rate.

Future work in this are should be directed at the enlargement of the scenarios of this study. In this work only straight roadways were considered. Further extensions of the dissemination algorithm should include road intersections as well as complex maneuvering of vehicles like possible change of direction or overtaking. In this work scenarios and vehicular traffic theory concepts were simplified in order to obtain a first evaluation of the possible performance of the algorithm.

References

1. Rudack, M., Meincke, M., Lott, M.: On the dynamics of ad hoc networks for inter-vehicle communications (ivc). Proc. of Int´l Conf. on Wireless Networks (ICWN´02) (2002)
2. Schnabel, W., Lohse, D.: Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung. Verlag für Bauwesen, Berlin, Bd. 1, 2 Auflage (1997)
3. Beyer, Hackel, Pieper, Tiedge: Wahrscheinlichkeitsrechnung und mathematische Statistik. Teubner Verlagsgesellschaft, (Leipzig, Germany)
4. Benz, T., Schäfers, L., Stiller, C., Vollmer, D.: Feasibility study on truck planning on european motorways. Deliverable D08.1 of ITS project PROMOTE-CHAUFFEUR (1999)
5. Leutzbach, W.: Introduction to the Theory of Traffic Flow. Springer, Berlin, (1988)
6. Gazis, D.C., Herman, R., Potts, R.B.: Car following theory of steady state traffic flow. Operations Research **7** (1959) 499–505
7. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Dynamical model of traffic congestion and numerical simulation. Physical Review **E 51** (1995) 1035–1042
8. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal de Physique I France **2** (1992) 2221–2229
9. Vollmer, D., Hiller, A.: Problemorientierte verkehrsmodellierung auf bundesautobahnen (2001)

4

Position-Based Routing for Car-to-Car Communication

Holger Füßler[1], Martin Mauve[2], Hannes Hartenstein[3], Christian Lochert[2], Dieter Vollmer[4], Dagmar Herrmann[4], and Walter Franz[4]

[1] University of Mannheim, Computer Science IV,
A5, 6 68159 Mannheim, Germany
fuessler@informatik.uni-mannheim.de

[2] University of Düsseldorf, Institute of Computer Science,
40225 Düsseldorf, Germany
{mauve,lochert}@cs.uni-duesseldorf.de

[3] University of Karlsruhe, Institute of Telematics,
76128 Karlsruhe, Germany
hartenstein@rz.uni-karlsruhe.de

[4] DaimlerChrysler AG, Stuttgart and Ulm,
{dieter.vollmer,dagmar.herrmann,walter.franz}@daimlerchrysler.com

Summary. Looking at the characteristics of vehicular traffic on highways and in cities, we discuss the benefits of position-based routing approaches over purely topology-based routing approaches for vehicular ad hoc networks (VANETs). We present detailed quantitative results of a NS-2-based simulation study for VANETs on German highways comparing greedy position-based routing with Dynamic Source Routing (DSR). For city scenarios, we outline some of the problems of position-based routing in dealing with radio obstacles like buildings. We propose a greedy perimeter coordinator routing and a method for detecting junctions based on the correlation coefficient of the neighboring vehicle's positions.

4.1 Introduction

Applications for car-to-car communication have heterogeneous requirements regarding the underlying communication technology. The transmission of warning messages in emergency situations, access to the Internet, and coordinated driving are just a few examples that illustrate this diversity. Any technology used for car-to-car communication must therefore be able to flexibly support the distinct communication patterns required by the individual applications. The general characteristics of mobile ad-hoc-networks match this requirement very well. Using the idea of mobile ad-hoc-networks each vehicle acts not only as an end-system by sending and receiving data. It also forwards data on behalf of other vehicles such that data can be routed from a sender to a receiver in a flexible manner without using a fixed infrastructure. Besides being flexible this approach also supports fast and direct communication between neighbouring vehicles as it is required for time critical emergency warnings. At the same time it can be offered without service fees, since no fixed infrastructure needs to be maintained by a service provider.

In this chapter we investigate how data can be routed through a network formed by cars. When compared to other mobile ad-hoc networks there are several important characteristics that are specific to car-to-car communication. On the one hand relative speeds between vehicles may be extremely high, in particular between vehicles driving in opposite directions. This leads to a very dynamic network topology and therefore challenging routing decisions. On the other hand the movement of cars is very regular: most of the time they follow a street or a highway with almost constant speed. We take these characteristics into account by using a detailed model for vehicular traffic to investigate the use of ad-hoc routing algorithms for the exchange of data between vehicles. The model includes elements such as vehicle characteristics (e.g., a car has a different movement pattern than a truck) and driver behaviour (e.g., when does a driver decide to change lanes). Models like this are used to determine the lifetime of parts of a vehicle, such as shock absorbers or turn signals. Thus, these models have to be very accurate. As an output highly realistic movement patterns are produced.

Based on the realistic movement patters we explore two common scenarios for inter vehicle communication: communication between vehicles driving on a highway and communication between vehicles that move within a network of city streets. Given that the width of a street is usually much smaller than the radio range, communication on a highway is inherently one dimensional. The destination is located either in front of the sender or somewhere behind it. As a result routing data from one vehicle to another can been seen as equivalent to flooding the highway between sender and receiver with this data. In contrast communication between vehicles in a city is two dimensional. At each junction between the sender and the receiver a decision has to be made about the street that the data should follow. Communication between vehicles in a city therefore amounts to 'flooding' the data along a path of streets that leads from the sender to the receiver.

For both scenarios we analyze the characteristics of the dynamic topology formed by the vehicles. In particular we look at network partitioning aspects and want to understand whether oncoming traffic needs to be used for the routing of packets. We employ the network simulator ns-2 [1] and the accompanying AdHockey tool [2] for this purpose. Our focus is then on studying the applicability of distinct routing strategies to vehicular ad-hoc networks by means of simulation with ns-2. The key questions we want to answer is whether the use of positional information in a routing approach provides significant benefits for this kind of network. In addition we examine how existing approaches should be adapted to support inter-vehicle communication.

The remainder of this chapter is structured as follows: we outline the model for vehicular movement in Section 4.2 where we also describe the generated movement patterns for both the highway and the city scenarios. Section 4.3 summarizes the basic concepts of routing in mobile ad-hoc-networks. The highway scenario is then investigated in detail in Section 4.4 while the city scenario is examined in Section 4.5. In Section 4.6 we generalize our observations to car-to-area and car-to-roadside communication. Section 4.7 concludes this chapter and gives an outlook to future work.

4.2 Simulation of Vehicular Traffic

Vehicular traffic simulations can be classified coarsely into *microscopic* and *macroscopic* approaches [3].

When following a macroscopic approach, one focuses on system parameters like *traffic density* (number of vehicles per kilometer per lane) or *traffic flow* (number of vehicles per hour crossing an intersection) in order to compute a road's capacity or the distribution of traffic on a network of roads. In general, from a macroscopic perspective vehicular traffic is viewed as a fluid compressible medium and, therefore, is modeled as a special derivation of the Navier-Stokes equations.

In contrast, with a microscopic approach the movement of *each* individual vehicle is determined. In order to generate vehicle movement patterns for ad hoc routing experiments one clearly has to follow a microscopic approach, since the position of each individual vehicle is needed. Nevertheless, one also has to take care that a microscopic simulation does not result in unrealistic macroscopic effects. As the vehicle movements are generated by a 'pre-process' and complexity is therefore a minor concern, we decided to use a *Driver Behavior Model* [4, 5] for the microscopic traffic simulation. Such a model not only takes the characteristics of the cars into account but it also includes a model of the driver's behavior, like lane changing and passing decisions, traffic regulation and traffic sign considerations, or decreasing speed in curves, to name only a few. Driver Behavior Models are known to be highly accurate and are therefore used by vehicle manufacturers, e.g., to determine the lifetime of certain parts of the car.

Fig. 4.1. A 500m highway segment with a traffic density of 6 vehicles per kilometer and lane taken from our generated movement scenario.

As a simulator we use the well validated DaimlerChrysler-internal driver behavior simulation tool called FARSI. This simulator is regularly employed to generate traffic simulations for the product development and evaluation of DaimlerChrysler. In particular FARSI simulations show realistic speeds, distances, and macroscopic properties like traffic flow and lane usage. Thus, FARSI guarantees that the vehicle movement patterns forming the basis of our experiments are as realistic as possible.

4.2.1 Highway Scenario

The simulated area in the highway scenario covers an area of 30 km length with two lanes per direction and with an average of 6 vehicles per kilometer and lane. Furthermore, the so-called 50%-desired speed parameter v_f (the parameter v_f splits the the population of vehicles into two halfs: the ones with a desired speed of at most v_f and the ones with a desired speed larger than v_f) is set to 130km/h. We assume that 15% of all vehicles are trucks. In FARSI the oncoming traffic is generated as a separate simulation for a single direction, i.e., both directions are independent. The positions of the vehicles are recorded every half a second together with current speed, lane identifier, and acceleration. From this file we generated our ns-2 movement file by taking a 200 seconds slice of the scenario.

The described scenario corresponds to weak day traffic on a German highway. In order to get an impression of the topology of a highway scenario with such a traffic density, a snapshot with realistic proportions for a highway segment of 500 m is given in Figure 4.1.

Since the topology and the topological changes over time are of utmost importance for our routing experiments, we present in the following some properties of the generated scenario with respect to the distribution of velocities and lane usage.

Figure 4.2 shows the distribution of the initial desired speeds for the simulation (we quantized speeds into 10km/h bins). The corresponding cumulative distribution shown in Figure 4.3 matches very well the cumulative distribution taken from a 'real' measurement from a German highway.

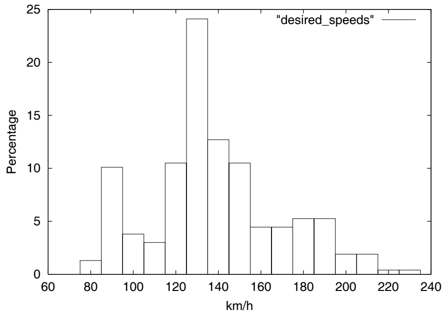


Fig. 4.2. Distribution of initial desired speeds.

The distribution of velocities in the simulated scenario with 6 vehicle per km and lane is given in Figure 4.4 (percentage of total time spent in a specific velocity class).

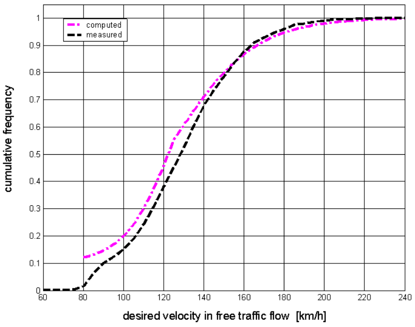


Fig. 4.3. Computed vs. measured cumulative distribution of initial desired speeds.

The lane usage measurement for our highway scenario shows 57.2% usage of the left lane and 42.8% usage of the right lane. This is typical for weak day traffic on a German highway since vehicles are only allowed to pass on the left lane.[5]

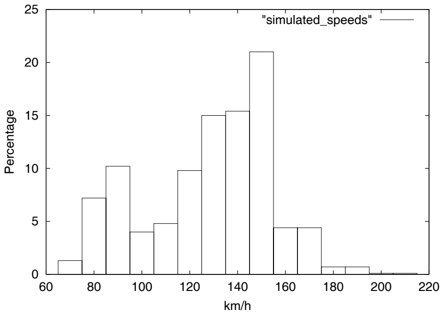


Fig. 4.4. Distribution of speeds in the simulated scenario with 6 vehicles per kilometer and lane.

General Observations

In order to get a first understanding of a vehicular (highway) ad-hoc network's topology and its dynamics we investigated the highway scenario from

[5] One has to note that speeds and lane usage depend on national regulations.

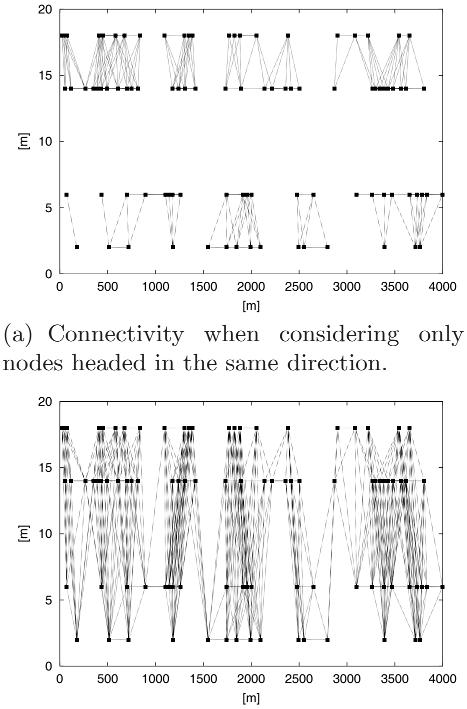
the previous section in a qualitative manner. Of particular interest was the theoretical connectivity of the ad-hoc network formed by the cars. One question we wanted to answer was whether or not it is necessary to route packets over oncoming traffic in order to get acceptable connectivity. This question is important since routing over oncoming traffic implies fast topological changes and potential problems on the physical level (doppler effect, etc.). As a simplification we first assumed that any two nodes can communicate when they are no more than 250 meters apart (approximating the behavior of IEEE 802.11). With the given average density of nodes (6 per lane per kilometer) network partitioning should then be very rare if the positions of the nodes were equally distributed.

In order to determine the connectivity we followed a designated node for 200 seconds on a 10 km path. For each node in a 3 km range of that node we calculated which other node could be reached directly and visualized this using Ad-Hockey. This was done twice: in the first experiment there was no communication allowed between vehicles driving in opposite directions. In the second experiment all vehicles on all four lanes were allowed to communicate with each other.

The result was converted to an MPEG video which can be downloaded from our web-server. A typical example of the connectivity when the directions are treated separately is given in Figure 5(a). This figure shows that both directions are partitioned. When investigating all 200 seconds of simulation time network partitionings are rather frequent, even though the average density of nodes is quite high. Clearly, the reason for this is that the position of vehicles is not equally distributed. This is caused by situations where one slow vehicle (e.g., a truck) overtakes another slow vehicle. In these situations connectivity will often break when oncoming traffic is not used to form the ad-hoc network.

In contrast Figure 5(b) shows the same situation when nodes on all four lanes are allowed to communicate with each other. It does show that partitionings of the network can be avoided by using oncoming traffic. An investigation of the full 200 seconds shows that most of the network partitionings can be alleviated in this fashion.

In addition we were interested in understanding how the amount of network partitions depends on the communication range. 4.6 shows the number of partitions on a 10 km segment with respect to the communication range of each individual node. Two graphs are given in the figure: the dotted one indicates the number of partitions when only vehicles driving in the same direction are considered for forwarding while the other graph describes the situation where all vehicles are taken into account. It can be seen that for the typical radio range of IEEE 802.11 (250 m) there are 7 partitions when only the vehicles driving in the same direction are taken into account. This is reduced to 2 partitions when all vehicles participate in the mobile ad-hoc network. Furthermore the graphs show that a communication range of 400 m would be desirable to completely eliminate partitioning in this scenario when



(a) Connectivity when considering only nodes headed in the same direction.

(b) Connectivity when considering nodes headed in both directions.

Fig. 4.5. Analysis of connectivity.

all vehicles are used or 1000 m if only vehicles driving in the same direction participate.

Based on these qualitative observations it seems likely that it will be necessary to route data packets over oncoming traffic even if the density of nodes headed in the same direction is quite high. If this is not done network partitionings can be frequent and each partitioning persists for a noticeable amount of time. Therefore an adequate technology for vehicular ad-hoc networks will have to support the routing of messages over oncoming traffic.

4.2.2 City Scenario

City traffic simulation itself is a complex challenge because the traffic flow in conurbation deeply depends on rules at its intersections and on the capacity of the roads and intersections. The traffic flow simulator Videlio [6], developed by DaimlerChrysler AG, extends Farsi and uses time depending origin destination matrices. Core elements are the Optimal Velocity Model [7], a

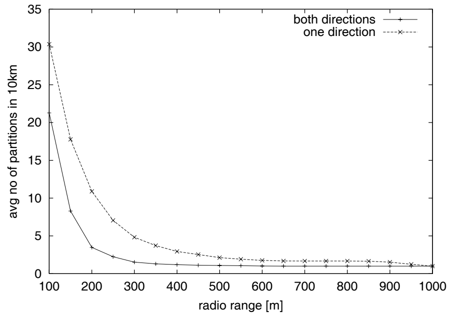


Fig. 4.6. Number of partitions with respect to radio range

special lane changing model [6] and the C-logit model [8] to calculate the traffic assignments. Videlio uses a detailed description of the road network with information about, e.g., lane numbers, traffic regulations, and time tables of the traffic lights.

For the vehicular movement pattern generation, a small part (6.25 km × 3.45 km) of the city of Berlin was modeled as a graph of streets with 28 vertices and 67 edges as depicted in 4.7. In total, the movement of 955 vehicles has been simulated.

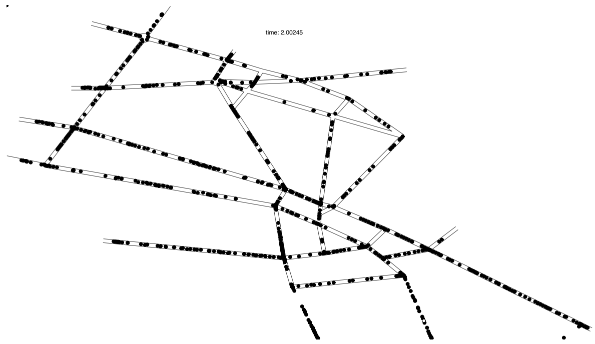


Fig. 4.7. Graph of streets of our vehicular movement simulations.

4.3 Unicast Routing in Mobile Ad-Hoc Networks

4.3.1 Protocol Classification

MANET research generated dozens of proposals for routing protocols, some of which are surveyed in [9, 10]). For this work we use a two-dimensional classification: topology-based vs. position-based protocols and proactive vs. reactive protocols.

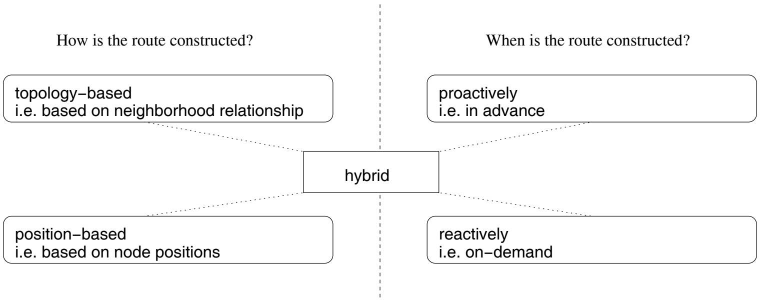

How is the route constructed?
When is the route constructed?
topology−based
i.e. based on neighborhood relationship
proactively
i.e. in advance
hybrid
position−based
i.e. based on node positions
reactively
i.e. on−demand

Fig. 4.8. Classification of MANET routing protocols

Fig. 4.8 depicts the structuring behind this classification. The first column divides by the question “How is the route constructed?”. Answering this question creates two classes of protocols. In the “based on topology” class, protocols build a route from source to destination based on the graph formed by the neighborhood relationships of nodes.

Position-Based protocols use the geographical positions of the forwarding node, its neighbors and the destination node to make forwarding decisions. These protocols do not guarantee to find the destination in all possible scenarios. Thus, they are complemented by so-called recovery strategies. In order to enable position-based protocols nodes need to be able to determine the position of the destination. This information can be provided by a “location service”, i.e. a distributed algorithm to determine the current geographic position of any given node.

The question “When is the route constructed?” is behind the second dimension of the classification. Here, we distinguish between protocols always acquiring routing information between all network nodes and those generating this information on-demand, i.e. only when a nodes want to communicate. Position-based protocols may contain proactive and/or reactive elements both at the level of the routing strategy and the location service.

Hybridity is—as the evolution of MANET protocols proceeds—a property a lot of modern protocols have in common. In this sense, protocols having their roots in different classes according to this classification use elements common to the other class to solve problem the other class is bad in. Thus, a convergence of algorithms can be observed.

4.3.2 Discussion and Selection of Candidate Protocols

One key attribute of Vehicular Ad-Hoc Networks is the high mobility of the involved nodes. While nodes in other Ad-Hoc Networks may move slow, the natural “movement mode” for a car ist at velocities higher than $10m/s$. To worsen this, almost half of the cars travel in the precise opposite direction, maximizing the relative speed. On the other hand, movements are usually highly predictable because cars commonly move on streets. Moreover, the street arrangement changes very slowly and could be integrated into a precomputed movement prediction scheme. In addition to that, a growing percentage of vehicles are aware of their position by means of a GPS-based navigation system. GPS also provide very accurate clock information and thus can be used to synchronize network nodes to a common time base which is necessary—or at least helpful—for a lot of network protocols.

Both, high mobility and the availability of position information let us to the assumption that position-based approaches would be an excellent choice for routing in VANETs. In addition position-information can get coarser, the further away an intermediate node is from the destination which is favorable in terms of scalability. Finally, a lot of Vehicular Applications seem to be highly position-oriented, i.e. the addressing will be rather position-based than by logical IDs as in terms of unicast or even multicast routing. Looking for a suitable position-based protocol, we selected GPSR [11]. GPSR uses greedy forwarding for position-based routing and provides a recovery strategy when greedy forwarding fails. It has been thoroughly tested and a ns-2 implementation of the protocol source code is freely availabel.

For topology-based protocols the Inter Engineering Task Force’s MANET group [12] has been active to standardize four protocols. These include two reactive protocols: Ad-Hoc On-Demand Distance Vector Routing (AODV [13]), and Dynamic Source Routing (DSR [14]), as well as two proactive protols, the Topology Dissemination Based on Reverse-Path Forwarding (TBRPF [15]) and the Optimized Link-State Routing Protocol (OLSR [16]). The proactive protocols, however, seemed to be inherently unsuitable for VANETS due to the extremely high overhead of keeping routes between all nodes in a network that changes its topology frequently. Therefore we selected AODV and DSR as topolgy-based approaches.

4.3.3 Candidate Protocol Description

Ad-Hoc On-Demand Distance-Vector Routing (AODV)

The Ad-Hoc On-Demand Distance Vector Routing is a topology-based routing method relying on distance vector tables well-known from the wired world, e.g. in the Routing Information Protocol (RIP) [17]. Fundamentally, the AODV “Path Discovery Process” separates into the “Reverse Path Setup”, building a route from the destination to the source node and the “Foward Path Setup” creating a route from the source to the destination. Due to its strictly reactive nature, a path discovery is initiated only when packets are to be transmitted by the sender.

In principle AODV works as follows: the node that wants to transmit a packet floods a route request. Whenever a node receives a route request packet, the node adds a route to the source node to its local distance-vector table consisiting of the hop distance to the request's source and the next-hop to be used (which ist the neighbor the request came from). If the packet is the requested destination, it also sends a reply packet (`RREP`) towards the source. By doing this, the "forward path setup" phase is entered. In this phase, the `RREP` travels back towards the source using the route set up in the reverse path setup phase. At every hop, the nodes add a route towards the destination to their table. When the `RREP` reaches the original source of the request, a bidirectional route is in place which can be used for data forwarding. (The exact protocol described in [13, 18])

In AODV the flooding is performed as an expanding ring search, i.e. at the beginning the route request is trasmitted with a small time to live. If there is no reply then the flooding is repeated with a larger time-to-live. This continues untill a preset maximum is reached or a reply is received. This technique is also called "adaptive flooding" and saves network resources as opposed to simply issuing a flood packet with the hop limit set to the maximal allowed network diameter.

Dynamic Source Routing (DSR)

Dynamic Source Routing (DSR) is typically performed in two steps: route discovery and route maintenance. A node that wants to send a packet to another node first checks its local route cache. This cache contains all valid routes the node knows about. If no route to the destination is present in the cache, a route discovery is performed. Essentially the route discovery requires that the node performing it floods the network with a *route request* that contains the ID of the node it wants to contact. Whenever the route request is forwarded by a node the forwarding node's ID is recorded in the packet. When it finally is received by the destination the route request contains a valid path from the source to the destination. The destination then sends a *route reply* back to the sender on the path contained in the route request. The sender and all nodes on the path from the destination to the sender put the route in their route cache. In order to reduce the amount of flooding, DSR employs a number of additional algorithms. For example intermediate nodes that have a valid route to the destination in their route cache may answer the route request directly with the information from the cache. In most situations these additional algorithms can prevent a full flooding of the network.

As long as two nodes communicate with each other *route maintenance* makes sure that a path between both nodes exists. When a path breaks, a packet that cannot be forwarded will generate a *route error* which is sent back to the sender of the original packet. On its way the route error causes the removal of the invalid route from the route caches of the intermediate nodes and of the sender. The sender then performs a new route discovery to find a new route to the destination. As with route discovery the route maintenance is supported by a number of additional algorithms that optimize the behavior of DSR. For a full description of DSR the reader is referred to [14].

Reactive Location Service

A location service is a requirement for most of the position-based routing algorithms known. Its basic task is to find the current geographic position of any node in the network which most position-based routing algorithms like GPSR [11] take for granted. When designing a real system or even only when performing a fair comparison, one has to fill this gap. The proposals found in literature where rather complex and seemed to be ill-suited for our purposes. Thus we have described and evaluated a straight-forward location service called "reacitve location service" (RLS) [19], that is inspired by the route discovery known from e.g. AODV or DSR: Whenever the position of a node is required, the node looking for position information floods a request containing the ID of the node it is looking for. The request contains the ID and position of the requesting node. When a node receives a request with its own ID, it replies to the node looking for its position.

In order to reduce the range of the flooding an expanding ring search is performed: the flooding starts with a range of 2 hops and is repeated with a greater range when no response is received during a certain time. The range of the flooding can be increased, e.g., linearly or exponentially.

With the reactive location service there is only overhead when data actually needs to be transmitted. This makes the comparison with reactive ad-hoc routing strategies quite fair. Using one of the existing location services would produce an overhead which does not (directly) relate to the transmitted payload data. Thus any results would depend on how much payload data is transmitted.

Clearly, the overhead of the reactive location service will be generally high if communication partners are changed frequently - and thus may be inferior to exiting approaches in those situations. However, for comparison purposes it seems to be more appropriate than proactive location services. In addition the reactive location service could be optimized, using caching and prediction of a node's future location based on its speed and heading. This was not part of this study and is left for future work.

Position-Based Greedy Routing with Beacons

The fundamental idea of position-based routing is rather old [20] and works as follows. Each node knows about its own geographic position. This position and the node's unique identifier is then broadcast as a so-called beacon packet after a fixed period of time. Whenever a node receives the beacon of another node, it adds the information it contains to a data structure called the neighbor table listing the last-known position of the respective neighbors. A second part of the algorithms expires the table's rows after a certain time interval in which no beacons of the corresponding entry where overheard.

After the neighbor table is obtained and a node `S` wants to send a data packet to a node `D`, it first uses a location service (see last section) to determine `Ds` current geographic position. Following the "Most-Forward-Progress within Radio Range" heuristic, `S` calculated for every neighbor in the neighbor table the distance progress the packet would make if this neighbor would be selected as a forwarder. After eliminating the nodes with negative progress (greedy behaviour), it selects the neighbor with the greatest progress as the next hop.

Recovery-Methods for 'non-greedy' situations

While a greedily-behaving algorithm is able to find shortest routes in dense networks, there are so-called void situations where this routing fails even if a route exists. In this case, so-called "recovery-strategies" were proposed to find routes in these cases. Most of these algorithms like the "Perimeter Mode" of GPSR [11] or face-2 [21] or newer enhanced versions like GOAFR [22] base on the distributed planarization of the network graph. On this planarized graph recovery is performed by applying variants of graph traversal algorithms like the left or right-hand rule. These strategies are proven to succeed under the assumption of static mobility and unit disk connectivity. However, mobility can seriously tamper the performance [23] and even if it works, there are cases where GPSR and face-2 would generate very long routes.

4.4 Forwarding in Highway Scenarios

In the following we analyze the quantitative behavior of DSR and GPSR/RLS when applied to a network of vehicles driving in a highway scenario.

4.4.1 Simulation Setup

The environment used for the simulation is based on the all-in-one distribution of ns-2.1b8a running under Linux. The `GPSR` code of Brad Karp was ported to this platform. The `DSR` code used is the one integrated in the distribution. We took a time slice of 200 seconds of the input data and a reduced kilometer range of 10 km (Position from 10 km to 20 km of the original data). This results in about 300 nodes in the scenario.

All experiments were conducted with two different MACs. One was IEEE 802.11 as provided in ns-2. The other one was an idealized MAC we implemented to abstract from MAC-specific effects. This *0-MAC* allows communication between two nodes if they are 250 meters or less apart and does not impose any upper limit on the amount of transmitted data. Collision between distinct packets that are simultaneously transmitted do not occur with the 0-MAC.

Communication Pattern

For the selection of the communication pattern we used the following algorithm. At any time there are 10 pairs of one sender and one receiver. These pairs are randomly selected such that they are no more than a maximum communication distance (in meters) from each other apart. In addition they are guaranteed to be theoretically able to reach each other during the time they communicate (i.e., they do not reside in different partitions). The sender then transmits 4 packets per second over a time of 5 seconds. The starting time is randomized in order to prevent synchronization. Whenever a message is successfully delivered, the receiver sends a reply. Thus we simulate typical bidirectional traffic as produced, e.g., by TCP. All packets carry a payload of 64 byte. The maximum distance between senders and receivers was varied from 500 meters to 4500 meters. Since the selection of partners is random (equally distributed) among the nodes fulfilling the constraints, sender and receiver can travel in the same or in different directions.

A Note about Border Effects

When simulating a linear street scenario, one has to consider border effects. For instance, a node leaving the studied area has to be deactivated, for its real position is off scope of the simulation. To accomplish that, we used the energy model of ns-2. If a node reaches the border of the simulated area, it is deactivated and reactivated (again) when it (re)-enters the scenario. In our scenario, since no node is allowed to travel backwards, each node is activated exactly and deactivated at most 1 time. Of course, when a node is deactivated, it stops sending `GPSR` beacons.

DSR Setup

The parameters originally set in the ns-2 implementation of `DSR` were kept for our simulation. The only modification was done to increase the maximum hop distance that a DSR route can span from 16 to 32 so that it is possible to reach all destinations even in the 4500 meter communication pattern. For a deeper understanding of `DSR` optimization, please refer to [14]. In our simulation `DSR` uses the promiscuous mode of the network interface to investigate all packets receivable regardless of the destination address.

PBR and RLS Setup

The Position-Based Routing algorithm we used is based on the code of GPSR [11], except the perimeter mode was turned off. The setup is as follows: the beacon information of a node, i.e. its own position, is piggybacked on every packet (data packets and location service packets) that it forwards. When piggybacking a beacon the node resets the timer for the scheduling of its next beacon. We varied the beacon interval between $\{0.25, 0.5, 1, 2\}$ seconds to study its influence on the rate of successfully delivered packet and routing overhead. We make use of the MAC callback feature, enabling a node to reroute packets still buffered by the MAC if a MAC link breaks. Although this is a violation of the strict layer separation, the gain of it is remarkable according to [11].

Our Reactive Location Service was used first with linear expanding ring search and then with exponential expanding ring search. The timeout value for triggering the flooding with an increased range was set to 100ms multiplied with the number of hops in the last cycle. The maximum hop-count for the flooding was set to 32, the same value used by DSR. This should enable us to reach any node in the simulated area. Each data packet and each reply sent in response to a data packet contains the ID and location of its sender and its receiver. Thus the location information about a communication partner is updated by the receipt of a packet from that communication partner.

4.4.2 Simulation Results

0-MAC

In order to gain an impression that is unaffected by the properties of the MAC we started the simulations by using the 0-MAC. The first experiments were conducted for the position based approach with a linear expanding ring search (increase of 1 hop per cycle). Surprisingly we had many cases where a destination node was not reached by the flooding. A more detailed analysis helped us to understand the reason for this: the problem occurs when two vehicles want to communicate which drive in different directions. For the first flooding a range of 2 was used while the vehicles were n hops apart (n was greater than 2). Flooding with range 2 therefore remained without success. However, during the time required for the first cycle to time out, the cars moved in opposing directions so that they now were at least $n + 1$ hops apart: the expanding ring search was slower than the vehicles. We concluded that for vehicle communication linear expanding ring search is not suitable as location service. Thus in the following we only consider exponentially expanding ring search.

One key performance metric for the suitability of a given approach is the rate of successfully delivered packets. Figure 4.9 shows this metric for DSR and GPSR with increasing maximum communication distances. There is just one plot for GPSR since all tested beaconing frequencies provided the same results in all ranges. This is no surprise since the flooding for the location service allows to piggy back the beacon information of all nodes between sender and receiver at the beginning of the communication. Furthermore the data packets sent will also be used for piggy backed beacons and keep the information about neighbors up-to-date. We tested beaconing frequencies with up to 16 seconds between beacons without major change in the outcome of the experiment.

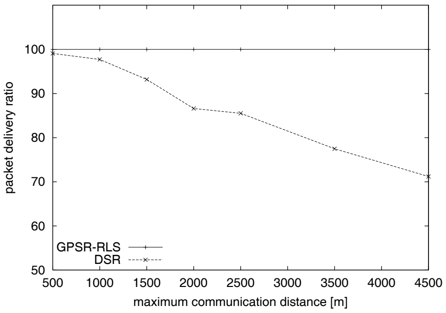
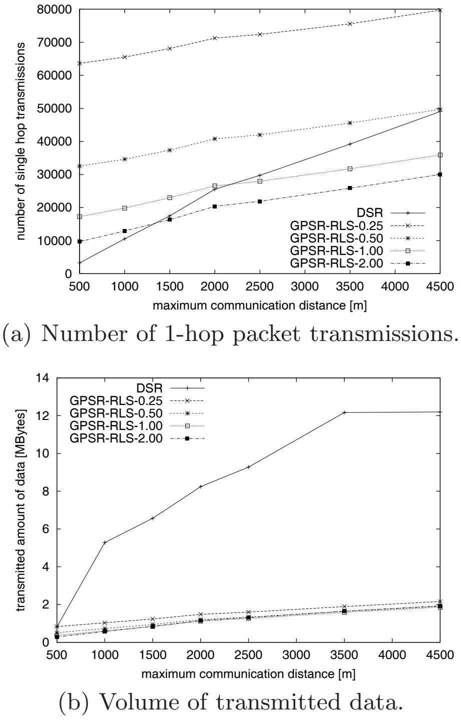


Fig. 4.9. Packet Delivery Ratio w.r.t. Maximum Communication Distance using the 0-MAC

Figure 4.9 can be interpreted as follows: as expected the rate of successfully delivered packets for DSR diminishes when the maximum communication distance becomes larger. This is caused by the fact the DSR needs to maintain a route from the sender to the receiver which becomes harder when the length of the route increases. The position based approach stays at the perfect packet delivery rate of 100% for all distances.[6] This can be explained by the properties of position based approaches: packet drops can occur only for one of the following three reasons: (1) if a local maximum is reached. This is extremely unlikely in our scenario. (2) If the information about the position of the local neighbors is inaccurate. Again this is very unlikely since the flooding of the location service in combination with piggy backed beacons will provide nearly perfect information about the neighbors. (3) If the information about the position of the destination is inaccurate. This is also very rare, since using the 0-MAC the reply containing the position of the destination requires only minimal time to reach the sender, thus it is very accurate when the data packet is transmitted.

Besides looking at the delivery rate it is also important to investigate how many packets and how much data is required to transmit a certain amount of payload data. We therefore measured the total number of one hop transmissions that occurred over the whole lifetime of the simulation. This is shown in Figure 10(a). Both unicast and broadcast messages are included in this figure. For GPSR we show the communication costs for all beaconing frequencies. It can be seen that the value for DSR starts low when the maximum communication distance is small and grows fast with increasing communication distances. This is caused by the increase in overhead for route establishment and maintenance which are the main sources of packets for DSR (besides the

[6] It should be emphasized that we did not try to "optimize" the simulation to achieve this figure. In fact we would have preferred a somewhat less perfect result. We invite people to validate these results and will therefore put up everything required to run the simulation on the web.



(a) Number of 1-hop packet transmissions.

(b) Volume of transmitted data.

Fig. 4.10. Analysis of communication costs.

actual data packets). GPSR/RLS on the other hand starts at a higher value and then increases more slowly. Furthermore it can be noticed that the communication overhead scales almost linearily with the beaconing frequency. The reason for this behavior is that beacons are the dominating source of one hop transmissions in position-based routing. These are independent of the maximum distance between communication partners. Since the packet delivery ratio is almost independent of the beaconing frequency and since beaconing provides the dominating amount of one hop packet transmission it seems appropriate to use a fairly low beaconing frequency when employing GPSR/RLS for vehicular networks.

Figure 10(b) displays the total amount of data used in form of single hop transmissions. It demonstrates that DSR needs significantly more data than GPSR for all examined maximum communication range values and beaconing frequencies. This is caused by the size of the packets needed to establish and maintain routes in DSR. Since these packets need to carry a source route from the sender to the receiver they can become quite large. As a contrast the packet size of GPSR/RLS is very small: all that is required is the position information and ID of the sender (and of the receiver if it is a data packet).

IEEE 802.11

In a second round we repeated the experiments using the default implementation of IEEE 802.11 in ns-2 as MAC. Given the results from the previous section we expected similar but somewhat less optimal results. In our initial experiments with IEEE 802.11 we were surprised to see that GPSR/RLS actually performed similar and sometimes worse than DSR in respect to the rate of successfully delivered packets. In particular the exponential expanding ring search frequently failed to reach the destination node. Investigating this problem we noticed that the flooded packets tended to synchronize themselves such that they cause collisions at the MAC layer. In IEEE 802.11 broadcast packets that are affected by such a collision are not retransmitted and remain lost. Thus the synchronized broadcasting of packets can lead to a complete whipeout of the affected packet. As a consequence we introduced a jitter when sending broadcast packets for the expanding ring search. This solved the problem.

Figure 4.11 shows the packet delivery rate for the simulation with IEEE 802.11. Generally the outcome is very similar to the 0-MAC case. However, there is one minor detail that is worth mentioning: for GPSR/RLS we had some runs where data packets got lost, even though the vast majority of runs did complete without a single packet loss. The main reason for those losses was that beacons and broadcast packets from the location service would still sometimes collide. Thus the information about the position and availability of neighbors is less accurate in the simulation runs with IEEE 802.11. This sometimes causes a forwarding node to be ignorant of the only neighbor with forward progress in the direction of the destination. The cost for the communication remained very similar to that of the 0-MAC case and is therefore not shown here.

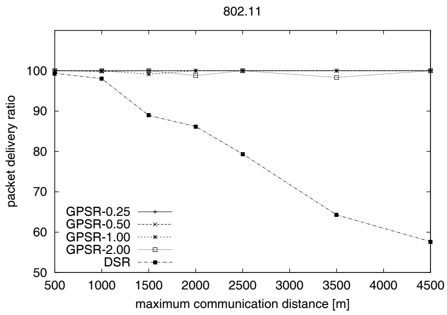


Fig. 4.11. Packet Delivery Ratio w.r.t. Maximum Communication Distance using IEEE 802.11

4.5 Forwarding in City Scenarios

Position-based routing with beacons such as in GreedyPerimeter Stateless Routing (GPSR) [11] is very well suited for highly dynamic environments such as inter-vehicle communication on highways. However, it has been discussed that radio obstacles [23], as they are found in urban areas, have a significant negative impact on the performance of position-based routing. In prior work [24] we presented a position-based approach which alleviates this problem and is able to find robust routes within city environments. It is related to the idea of position-based source routing as proposed in [25] for terminode routing. The algorithm needs global knowledge of the city topology as it is provided by a static street map. Given this information the sender determines the junctions that have to be traversed by the packet using the Dijkstra shortest path algorithm. Forwarding between junctions is then done in a position-based fashion. In this short paper we show how position-based routing can be aplied to a city scenario without assuming that nodes have access to a static street map and without using source routing. (This section is a modified version of [26]).

4.5.1 Position-based routing

In existing position-based routing approaches an intermediate node forwards a packet to the direct neighbor which is closest to the geographic position of the destination. This is called greedy forwarding. For this task each node has to be aware of *i)* its own position, *ii)* the position of its direct neighbors and *iii)* the position of the final destination. A node determines its own position by using GPS, the position of the neighbors is received through one hop beacon messages transmitted periodically by all nodes and the position of the final destination is provided by a location service [27] or by a geocast application. Since greedy forwarding uses only local information a packet may reach a local optimum w.r.t. the distance to the destination, i.e. no neighbor exists which is closer to the destination than the intermediate node itself. In order to escape from a local optimum a repair strategy may be used. The general aim of a repair strategy is to forward the packet to a node which is closer to the destination than the node where the packet encountered the local optimum. Once such a node is reached greedy forwarding can be resumed. Several repair strategies have been proposed, including Greedy Perimeter Stateless Routing [11] and face-2 [28]. However, it has been shown [23, 24] that existing repair strategies do not perform well in city environments because they rely on distributed algorithms for planarizing graphs. In the presence of radio obstacles the use of these algorithms frequently partitions an otherwise connected graph, making the delivery of packets impossible. As a result we propose a new routing approach for mobile Ad-Hoc Networks which we call Greedy Perimeter Coordinator Routing (GPCR).

4.5.2 Greedy Perimeter Coordinator Routing

Greedy Perimeter Coordinator Routing (GPCR) is a position-based routing protocol. The main idea of GPCR is to take advantage of the fact that streets and junctions form a natural planar graph, without using any global or external information such as a static street map. GPCR consists of two parts: a restricted greedy forwarding procedure and a repair strategy which is based on the topology of real-world streets and junctions and hence does not require a graph planarization algorithm.

Restricted Greedy Routing

As long as no global optimum is encountered, a special form of greedy forwarding is used to forward a data packet towards the destination. Since obstacles (e.g., buildings) block radio signals, data packets should be routed along streets. Junctions are the only places where actual routing decision are taken. Therefore packets should always be forwarded to a node on a junction rather than beeing forwarded accross a junction. This is illustrated in Figure 4.12 where node u would forward the packet beyond the junction to node $1a$ if regular greedy forwarding is used. By forwarding the packet to node $2a$ an alternative path to the destination node can be found without getting stuck in a local optimum. In the remainder of this work we call nodes that are located in the area of a junction a *coordinator*. A coordinator broadcasts its role along with its position information. In a first step we assume that each node knows whether it is a coordinator (i.e., located in the area of a junction) or not. We will show in section 4.5.3 how a node can learn about this information.

If the forwarding node is located on a street and not on a junction the packet is forwarded along the street towards the next junction. To achieve this, the forwarding node selects those neighbors whose positions approximate an extension of the line between the forwarding node's predecessor and the forwarding node itself. Out of these qualified neighbors one has to be selected as the next hop of the packet. As long as there are no qualified neighbors which are coordinators the node with the largest distance to the forwarding node is chosen. If coordinators are qualified then one coordinator is randomly chosen as the next hop. With this approach packets will not be forwarded across junctions. Figure 4.13 shows an example of how the next hop is selected on a street. Node a receives a packet from node b. Because a is located on a street and not on a junction it should forward the packet along this street. First the qualified neighbors of a are determined. Then it is checked whether at least one of them is a coordinator. As in this example there are three coordinator nodes that qualify as a next hop one of these coordinator nodes is chosen randomly and the packet will be forwarded to this coordinator.

Once a packet reaches a coordinator a decision has to be made about the street that the packet should follow. This is done in a greedy fashion: the neighboring node with the largest progress towards the destination is chosen. This implies a decision on the street that the packet should follow.

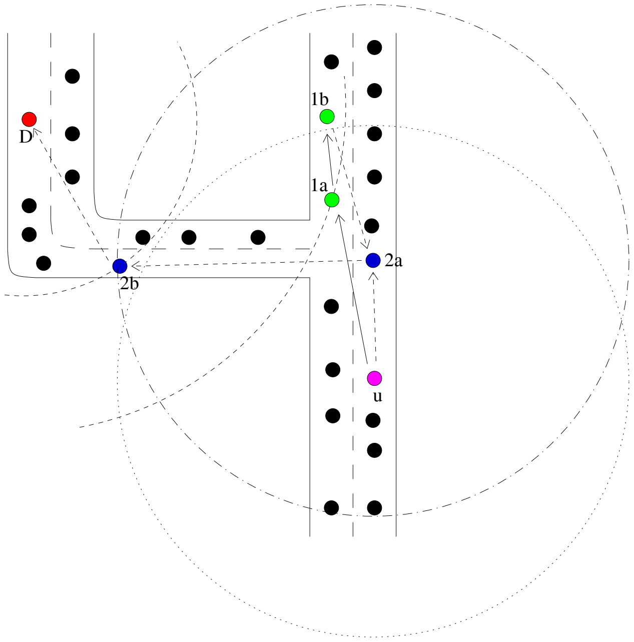


Fig. 4.12. Greedy Routing vs. Restricted Greedy Routing in the area of a junction.

Repair Strategy

Despite of the improved greedy routing strategy the risk remains that a packet gets stuck in a local optimum. Hence a repair strategy is required. The repair strategy of GPCR avoids using graph planarization by making routing decision on the basis of streets and junctions instead of individual nodes and their connectivity (which do not form a natural planar graph). As a consequence the repair strategy of GPCR consists of two parts: (1) On each junction it has to be decided which street the packet should follow next. (2) In between junctions greedy routing to the next junction, as described above, can be used.

If the forwarding node for a packet in repair mode is located on a junction (i.e., it is a coordinator) then the node needs to determine which street the packet should follow next. To this end the topology of the city is regarded as a planar graph and the well known right-hand rule [11, 28] is applied.

We illustrate the use of the right hand rule in figure 4.14. A packet with destination D reaches a local optimum at node S. The forwarding of the packet is then switched to the repair strategy and it is routed along the the street until it hits the first coordinator node. Node C_1 receives the packet and has to decide on the street the packet should follow. Using the right-hand rule it chooses the street that is the next one counter-clock wise from the street the packet has arrived on. Therefore node I will be chosen to forward the packet. The packet will then be forwarded along the street until the next junction

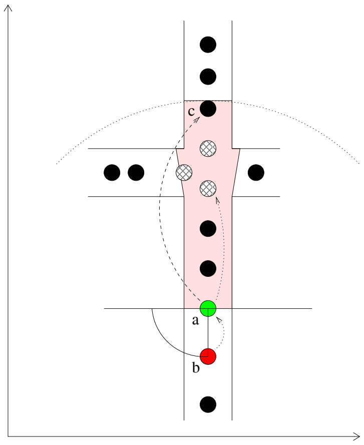


Fig. 4.13. Coordinator nodes are preferred to non-coordinator nodes.

is reached. When the packet arrives at the coordinator C_2 this node has to decide again on the next street that is to be taken and decides to forward the packet to node L. At this point the distance to the destination is less than at the beginning of the repair strategy at node S. Hence the mode is switched back to the greedy strategy described above.

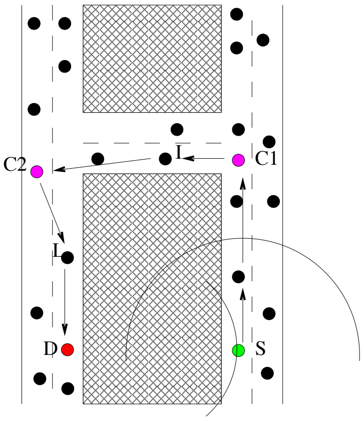


Fig. 4.14. The right hand rule is used on the level of streets as a repair strategy in GPCR.

4.5.3 Detecting junctions

One key challenge of GPCR is to detect whether a node is located on a junction without using external information. In the following we present two alternative approaches.

In the first approach each node regularly transmits beacon messages including the position of the node that is sending the beacon as well as the position of all of its neighbors. By observing the beacon messages a node has the following information for each neighbor: its position and the position and presence of the neighbor's neighbors. A node x is then considered to be located in a junction if it has two neighbors y and z that are within transmission range to each other but do not list each other as neighbors. This indicates that those neighbors are separated by an obstacle and that x is able to forward messages around this obstacle.

The second approach does not require special beacon messages. Each node calculates the correlation coefficient with respect to the position of its neighbors. We define x_i and y_i as the x-coordinate and y-coordinate of a node i. The variables x and y subsume the population of all these positions x_i and y_i, respectively. The mean of a population x is marked by $\bar{x}$. σ_{xy} indicates the covariance of two populations x and y and σ_x indicates the standard deviation of a population x. The correlation coefficient ρ_{xy} is then defined as:

$$\rho_{xy} = \left| \frac{\sigma_{xy}}{\sigma_x \sigma_y} \right| = \left| \frac{\sum\limits_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\sum\limits_{i=1}^{n} (x_i - \bar{x})^2 \right) \left(\sum\limits_{i=1}^{n} (y_i - \bar{y})^2 \right)}} \right|$$

with $\rho_{xy} \in [0, 1]$. A correlation coefficient close to 1 indicates a linear coherence as it is found when the node is located in the middle of a street. A correlation coefficient close to 0 shows that there is no linear relationship between the positions of the neighbors. Consequentially we conclude that the node is located on a junction. By adjusting a threshold ϵ a node can evaluate the correlation coefficient and assume with $\rho_{xy} \geq \epsilon$ that it is located on a street and with $\rho_{xy} < \epsilon$ that it is located within the area of a junction. We use a very large value $\epsilon = 0.9$ for our implementation to account for the highly linear relationship between node positions on streets.

4.5.4 Simulation Results

We simulated the performance of GPCR with the `ns-2` simulator version ns-2.1b9a. For the simulations we used a real city topology which is a part of Berlin, Germany. The scenario consists of 955 cars (nodes) on 33 streets in an area of 6.25 km × 3.45 km. The movement of the nodes was generated with a dedicated vehicular traffic simulator and represents a real world movement pattern for this given scenario [24]. IEEE 802.11 was used as MAC with a transmission rate of 2 Mbps. The transmission range was set to 500 m. Real world tests with cars have shown this to be a reasonable value when using external antennas. For each simulation run we randomly selected ten sender-receiver pairs. Each pair exchanges 20 packets over 5 seconds. We measured the achieved packet delivery rate (Fig. 4.15) versus the distance between the two communication partners and the number of hops (Fig. 4.16). The communication distance between two nodes is calculated as the minimal distance based on the street topology at the beginning of the communication. Each point in the graphs is based on 10 independent simulation runs.

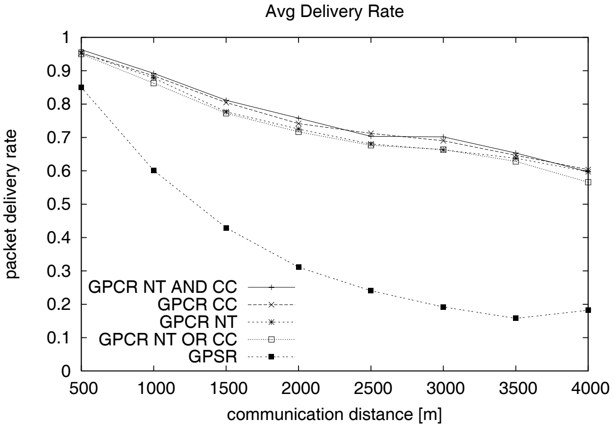


Fig. 4.15. GPCR vs. GPSR. – Delivery rate

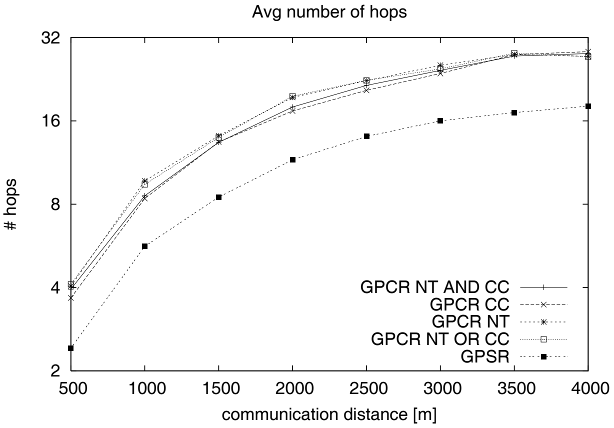


Fig. 4.16. GPCR vs. GPSR. – Avg. no. of hops

Fig. 4.15 also depicts how the delivery rate is influenced by the algorithms used for junction detection. It shows that calculating the correlation (CC) coefficient performs slightly better than relying on the comparison of the neighbortables of the neighbors (NT). We also analyzed a compound decision consisting of the neighbortable comparison and correlation coefficient, concatenated by logical OR as well as by logical AND. The latter one outperforms the other approaches slightly but it does not come for free: the size of the beacon packets increases for each of the two approaches. Therefore, GPCR simply uses the correlation coefficient. In general the study on achievable packet delivery rate (Fig. 4.15) shows good results for our approach compared to GPSR. This improvement in performance comes at the expense of a higher average number of hops and a slight increase in latency. This increase in hop counts and latency is mainly caused by those packets that could not be delivered at all by GPSR and thus did not impact the hop-count and latency for GPSR.

4.6 Generalization of Unicast Forwarding to Other Forwarding Modes

An interesting observation is the convergence of unicast, geocast and flooding in highway scenarios.[7] Assuming all nodes are able to listen to communication not originally destined for themselves (promisicious mode), unicast between two nodes is similar to flooding or geocasting to the highway segment between them. This allows for very efficient flooding algorithms.

Geocast is usually defined as addressing all nodes in a geographic region defined by a geometric shape. For street-bound car traffic, this region is the intersection of the geometric shape and the streets themselves. Street-based position encoding allows applications to address these streets directly. This can be highly desirable, for example when a safety application wants to let all cars traveling behind know that something dangerous happened. With street-based position encoding, limiting the area of information forwarding to a street comes natural whereas standard geocast requires that the street geometry itself is transformed into a geometric shape.

4.7 Conclusions and Future Work

In this chapter, we have subdiveded the problem of datagram packet forwarding in VANETs into *packet forwarding on highways* and *packet forwarding in city scenarios* and showed the specific challenges of both cases.

[7] Geocast ist the addressing of a geographic region and flooding is the addressing of all nodes, often within a certain hop-range.

For packet forwarding on highways, we have shown that position-based routing using beacons is well-suited to fullfill the task. In addition, we have observed that when a packet is fowarded on a highway, all vehicles are likely to overhear the packet. Thus, unicast fowarding, geocast forwarding and even flooding might converge in this scenario.

For the city scenarios, we have presented a new position-based routing approach, GPCR, which is able to deal with the challenges of city scenarios where obstacles often block radio signals. Our approach does not require external information such as a static street map to avoid the problems that existing position-based approaches face in this type of environment. Currently, we are looking into forwarding algorithms that make use of a static map. Under this assumption streets can be treated as links and junctions as nodes. A junction-to-junction beaconing protocol collects connectivity information and position-based forwarding is used both to reach the next junction and to decide, which street the packet should take.

Recent development in Vehicular Ad-Hoc Networks show that the focus is shifting from ent-to-end packet transport to geocast applications and from packet to information forwarding. This leads to new types of protocols and even a different view on protocol architectures [29].

References

1. NS-2 (The ns-2 network simulator) http://www.isi.edu/nsnam/ns/.
2. CMU wireless extensions. (The cmu monarch wireless and mobility extensions to ns-2) http://www.monarch.cs.cmu.edu/cmu-ns.html.
3. Helbing, D.: Traffic and related self-driven many-particle systems. Rev. Modern Physics **73** (2001) 1067–1141
4. Vollmer, D., Balasubramanian, B., Siegert, E.: Fahrtsimulation unter realistischen Umfeldbedingungen (in German). VDI-Berichte (1992)
5. Benz, T., Schäfers, L., Stiller, C., Vollmer, D.: Feasibility study on truck planning on european motorways. Deliverable D08.1 of ITS project PROMOTE-CHAUFFEUR (1999)
6. Kronjäger, W., Hermann, D.: Travel time estimation on the base of microscopic traffic flow simulation. ITS World Congress (1999)
7. Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y.: Dynamical model of traffic congestion and numerical simulation. Physical Review **E 51** (1995) 1035–1042
8. Zurheide, F.: Dynamische Verkehrsumlegung in einer mikroskopischen Simulati on. Master's thesis, Fachhochschule Braunschweig/Wolfenbüttel (1999)
9. Royer, E.M., Toh, C.K.: A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. IEEE Personal Communications (1999) 46–55
10. Mauve, M., Widmer, J., Hartenstein, H.: A Survey on Position-Based Routing in Mobile Ad-Hoc Networks. IEEE Network **15** (2001) 30–39
11. Karp, B.N., Kung, H.T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '00), Boston, Massachusetts (2000) 243–254

12. IETF MANET. IETF Mobile Ad-hoc Networks (manet) group. (http://www.ietf.org/html.charters/manet-charter.html)
13. Perkins, C.E., Royer, E.M.: Ad-Hoc On-Demand Distance Vector Routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA (1999) 90–100
14. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski, T., Korth, H., eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996) 153–181
15. Ogier, R.G., Templin, F.L., Bellur, B., Lewis, M.G.: Topology Broadcast Based on Reverse-Path Forwarding (TBRPF). Internet Draft, draft-ietf-manet-tbrpf-03.txt, work in progress (2001)
16. Clausen, T., Jacquet, P., Laouiti, A., Minet, P., Muhlethaler, P., Quayyum, A., Viennot, L.: Optimized Link State Routing Protocol. Internet Draft, draft-ietf-manet-olsr-05.txt, work in progress (2001)
17. Hedrick, C.L.: RFC 1058: Routing information protocol (1988)
18. Perkins, C.E., Belding-Royer, E.M., Das, S.R.: Ad hoc On-Demand Distance Vector (AODV) Routing. IETF RFC 3561 (Experimental) (2003)
19. Käsemann, M., Füßler, H., Hartenstein, H., Mauve, M.: A Reactive Location Service for Mobile Ad Hoc Networks. Technical Report TR-02-014, Department of Computer Science, University of Mannheim (2002)
20. Finn, G.G.: Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, ISI (1987)
21. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc Wireless Networks. In: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications (DIAL-M '99), Seattle, WS (1999) 48–55
22. Kuhn, F., Wattenhofer, R., Zollinger, A.: Worst-Case optimal and average-case efficient geometric ad-hoc routing. In: Proceedings of the fourth ACM international symposium on Mobile and ad hoc networking & computing (MobiHoc '03), Annapolis, Maryland (2003) 267–278
23. Karp, B.N.: Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning. Talk at DIMACS Workshop on Pervasive Networking (2001)
24. Lochert, C., Hartenstein, H., Tian, J., Füßler, H., Hermann, D., Mauve, M.: A Routing Strategy for Vehicular Ad-Hoc Networks in City Environments. In: Proc. of IEEE IV'03, Columbus, OH (2003) 156–161
25. Blažević, L., Giordano, S., LeBoudec, J.Y.: Self Organized Terminode Routing. Cluster Computing Journal **5** (2002)
26. Lochert, C., Füßler, H., Mauve, M., Hartenstein, H.: Geographic Routing in City Scenarios. ACM SIGMOBILE Mobile Computing and Communications Review (MC2R) **9** (2005)
27. Camp, T., Boleng, J., Wilcox, L.: Location Information Services in Mobile Ad Hoc Networks. In: Proceedings of the IEEE International Conference on Communications (ICC), New York City, New York (2002) 3318–3324
28. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks **7** (2001) 609–616
29. Füßler, H., Torrent-Moreno, M., Transier, M., Festag, A., Hartenstein, H.: Thoughts on a Protocol Architecture for Vehicular Ad-Hoc Networks. In: 2nd International Workshop on Intelligent Transportation, Hamburg, Germany (2005) 41–45

5

A Position-Based Router: Design, Implementation and Measurements

Holger Füßler[1], Michael Möske[1], Hannes Hartenstein[2], Walter Franz[3], Andreas Festag[4], and Christian Wagner[3]

[1] University of Mannheim, Computer Science IV,
A5, 6 68159 Mannheim, Germany
fuessler@informatik.uni-mannheim.de

[2] University of Karlsruhe, Institute of Telematics,
76128 Karlsruhe, Germany
hartenstein@rz.uni-karlsruhe.de

[3] DaimlerChrysler AG, Research & Technology Ulm,
89081 Ulm, Germany
walter.franz@daimlerchrysler.com

[4] NEC Europe Ltd., Network Labs Europe Heidelberg,
69115 Heidelberg, Germany
festag@netlab.nec.de

Summary. We describe the setup of a real-world vehicular ad hoc network that makes use of *i)* IEEE 802.11a/b hardware and of *ii)* position-based routing and forwarding mechanisms. We outline the rationale of the design, provide some implementation details and present some measurements for static and dynamic scenarios with up to four vehicles.

5.1 Introduction

In the framework of the FleetNet project, a demonstrator system was designed and implemented with the capability of demonstrating selected research achievements. NEC Europe Network Labs, the University of Mannheim, and DaimlerChrysler joined forces to provide the software part of the communication system, to integrate the communication system into a fleet of vehicles, and to provide basic demonstrable applications. A real world vehicular ad hoc network was set up that makes use of position information and that communicates by means of off-the-shelf 802.11a/b hardware.

In this chapter, we describe some efforts and results made in the process of building this system. While many research papers have their focus on 'results' rather than on describing how these results were achieved, we also like to outline how our real-world ad hoc network testbed was implemented and set up. During design and implementation, our key interest was on getting

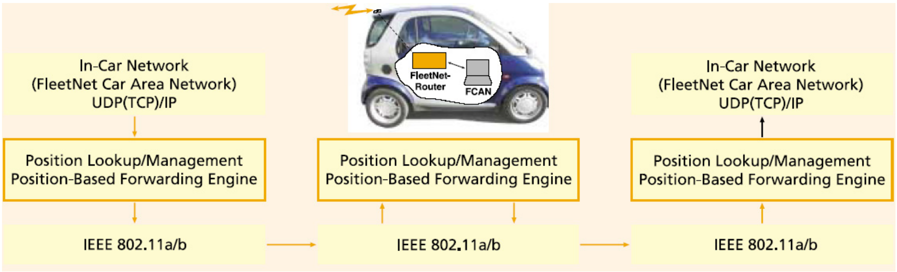

In-Car Network
(FleetNet Car Area Network)
UDP(TCP)/IP
Position Lookup/Management
Position-Based Forwarding Engine
IEEE 802.11a/b
FleetNet-Router
FCAN
Position Lookup/Management
Position-Based Forwarding Engine
IEEE 802.11a/b
In-Car Network
(FleetNet Car Area Network)
UDP(TCP)/IP
Position Lookup/Management
Position-Based Forwarding Engine
IEEE 802.11a/b

Fig. 5.1. Demonstrator overview

some first real-world experience with vehicular ad hoc networks in the field. The corresponding lessons learned provide feedback for further research and development work.

Section 5.2 describes the demonstrator system from a bird's eye perspective. In Section 5.3 we give some details on a how the node architecture is actually implemented and on how position information is encoded in the demonstrator system. The following Section 5.4 deals with the protocol for position-based packet routing and forwarding deployed in the demonstrator. After inital testing and debugging, we have performed some measurements that are described in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.2 The Demonstrator System: Overview

The FleetNet demonstrator consists of nine Smart™ cars, each car representing an IPv4 subnet connected to the others through a wireless mobile ad hoc network. Each vehicle is equipped with a Windows-based application PC representing the FleetNet Car Area Network (FCAN) and a Linux-based router (see Figure 5.1). The router is connected to the application machine via Ethernet while the connectivity to the routers of other nodes is provided by IEEE 802.11b. For a more illustrative description see [1]. Position-based greedy forwarding is implemented as routing protocol [2].

Since there is no immediate possibility to include position information into the IPv4 header, a layer 2.5 architecture was chosen: all data needed for routing is stored in the FleetNet routing header located between MAC and IP header. With a beacon-based position-based routing approach, each node has to have information on its own position, the position of its single hop neighbors as well as on the position of the destination node. To get its own position, the router is connected to the on-board navigation system (with integrated GPS). This position information is distributed among other nodes in radio range through beacon messages sent out periodically. Furthermore, all data packets include the position information on the sender to provide piggybacked beaconing. With greedy position-based forwarding, the router determines as next hop the neighbor that is closest to the destination. The destination's position is provided by a location service; in our case we make use of the Reactive Location Service (RLS) [3], which is based on a simple flooding mechanism to determine the position of a requested node. Location information can also be extracted from the FleetNet header of received data packets. A more thorough treatment of the issue of position-based routing and forwarding is given in Chapter 4. In this section we look at the demonstrator from the hardware and network architecture perspective.

Hardware Perspective

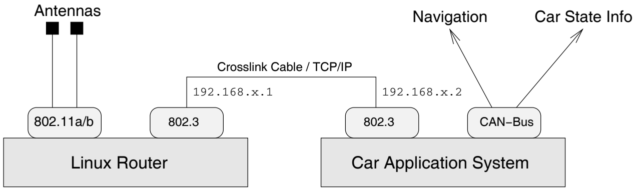


Fig. 5.2. Hardware Component Diagram

Figure 5.2 shows a more detailed view of a the communication equipment of a single car, i.e., of the various interfaces of the Linux-based router and the car application system acting as a gateway to the in-car network (or FleetNet car area network (FCAN)). Router and car application system are connected by a 100BaseT Ethernet. Since each of the cars is currently equipped with a single car application system only, the Ethernet network is collapsed to a single cross-link ethernet cable. However, the point-to-point connection could be easily extended to multiple car application systems by means of Ethernet hubs or switches. For communication on this network, IPv4 [4] is used. In order to support more than a single car application system, most of the interfaces to the routing system are IP-enabled, i.e. they can be accessed from outside the router system. The car application systems connects to the GPS-receiver and/or navigation system and to the CAN-bus (Controller Area Network) of the car.

The router uses standard IEEE 802.11 [5] *pcmcia/pc-card* hardware for radio communication. Both, 802.11a and 802.11b can be supported but major parts of the router software are optimized for the use of Lucent Orinoco wireless LAN network interface cards and the Linux driver software for these cards. Therefore, the usage of these optimizations is restricted to IEEE 802.11b (2.4GHz) network interface cards. Nevertheless, the system works with different drivers when the basic functionality of the Linux wireless tools is supported. To enhance radio connectivity, both radio systems are connected to passive external omni-directional antennae (gain of 4dBi [6]) mounted on the roof of the car (with different antennae for 8022.11 a and b).

Protocol Stack Perspective

Figure 3(a) shows the original proposal for the protocol stack of a node. In this architecture, two types of applications exist. The first type of applications is called FleetNet-unaware (FNU) whereas the other one is referred to as FleetNet-aware (FNA). FNU applications are basically standard IP-applications. Since they should behave as in any other system with Internet access, the FleetNet-specific functionality is hidden from them. The interface between these applications and the routing sub-system are standard IP sockets [7].

FleetNet aware applications, the second application type, are regarded to be more important for inter-vehicle communications. These applications are aware of communicating over an inter-vehicular network using position-based routing for packet forwarding, and are capable to exploit the corresponding specific features.

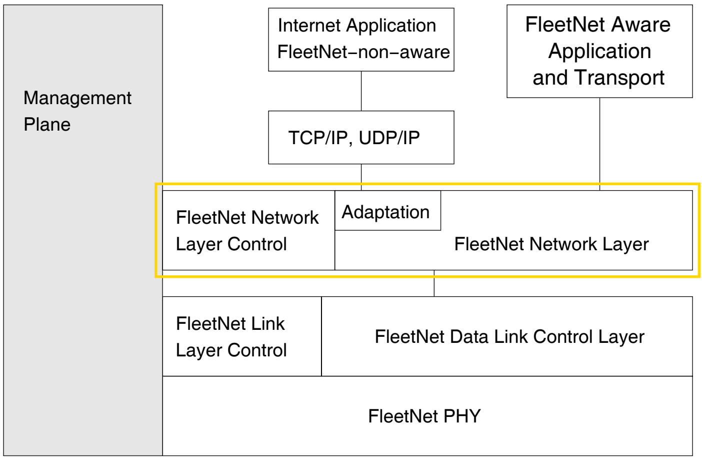
Figure 3(b) is a simplified version of the protocol layers as it is used in the demonstrator system. The lower protocol layers as depicted in Figure 3(a) are realized in hardware (network interface card) and the driver software, in the Linux operating system typically implemented as a *kernel module.*[5]. Due to a lack of a reasonable transport protocol for a VANET scenario, only a rudimentary transport functionality is provided in the demonstrator system that is essentially only used for application port (de)multiplexing (like in [8]). Therefore, basic transport interface functionality for FNA applications is provided by the routing system[6]. This fact is indicated by the extension of the box denoting the functionality scope of the FleetNet Demonstrator router. FNA applications directly interact with this interface for data transport. An additional interface is provided for accessing management information like router state.

To allow for applications to reside on a different computing devices other than the router, all interfaces above the box are implemented using UDP/IP. While the router transparently handles FNU packets, the FNA data forwarding / management interfaces are directly addressed by using UDP to a specific port.

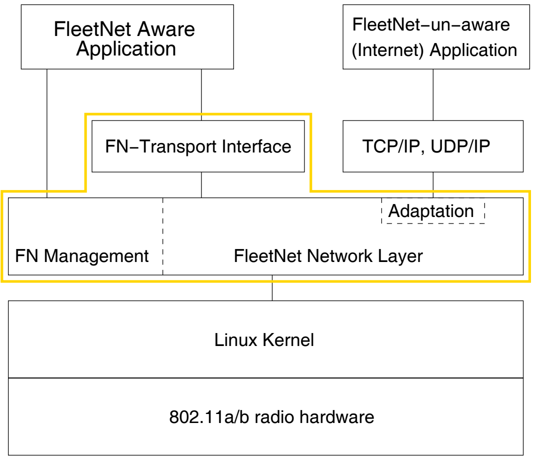
For IP addressing a simple and not scalable solution was used. As outlined above, only IPv4 [4] is used and the (yet) private nature of the network implies the usage of a reserved private address space [11]. The `192.168.x.y` address partition we chose for the router is subdivided in the following manner: The third byte of the address denotes the car, i.e. all cars have a different `x` byte. All systems in the same car or – to be more precise – all network interfaces of the in-car IP systems differ in the last byte `y`. In subnet-mask notation

[5] For simplicity, the interface of the driver is depicted by only one line, although one might also draw a more complex picture distinguishing between data forwarding and network management as it is actually used.

[6] Compared to TCP [9, 10], the transport functionality is only rudimentary, missing support for e.g. reliable end-2-end transport.


Management Plane
Internet Application FleetNet−non−aware
FleetNet Aware Application and Transport
TCP/IP, UDP/IP
FleetNet Network Layer Control
Adaptation
FleetNet Network Layer
FleetNet Link Layer Control
FleetNet Data Link Control Layer
FleetNet PHY

(a) FleetNet Network Architecture



(b) FleetNet Demonstrator Architecture (simplified)

Fig. 5.3. FleetNet Network Architecture

the FleetNet Demonstrator is thus a collection of `192.168.x.0/24` class C subnets interconnected by the FleetNet network acting as an intelligent link layer completely hiding its (potential) multi-hop property. For our convenience, the in-car ethernet interface of the router always has `1` as the `y` byte. All other systems use increasing `y` bytes. Since we usually have only a single in-car system, its IP address is `192.168.x.2` (see also Figure 5.2).

5.3 Implementation Issues

5.3.1 Vertical Communication

The protocol described in Section 5.4 is implemented in ANSI C for the Linux operating system as a forking user space software daemon. The implementation does not include any modifications of the Linux kernel. In order to execute the routing daemon, only a few kernel settings, including iptables rules and network interface settings, need to be defined.

The implementation design is illustrated in Figure 5.4: The router is equipped with two interfaces – a wireless interface (IEEE 802.11) and a wired interface (Ethernet 100BaseT). The protocol stack in the kernel space consists of the drivers for the interfaces and TCP/UDP on top of IP layers. The routing daemon is executed in the user space and accesses the kernel space by means of five different interfaces marked by numbers in Fig. 5.4.[7] The arrows between the interfaces and the router show whether packets are received from the interface, or sent via the interface or both.

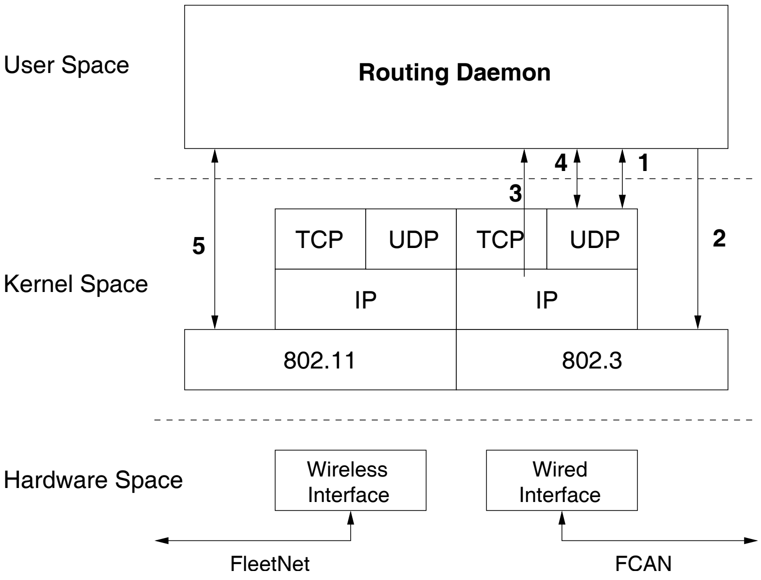


Fig. 5.4. Design of the Router and its Interfaces

In the following the implementation and purpose of the interfaces are briefly described.

[7] In the following, the term interfaces does not mean the network interfaces, but the different ways of sending and receiving packets of the router.

Management interface (1) The management interface is implemented as a UDP socket listening to default port 1501. The purpose of the interface is the data exchange between applications executed in the car application system and the router, more precisely to exchange location information to update internal data structures in the router as well as in the car application system. The data exchange includes the location updates sent from the car application system to the router, the triggering of a location query by the car application system, as well as the transfer of the *location table* from router to the car application system on a request/reply basis.

Car interface (2) is the interface used to send packets from the router to the car application system. It is noted that the router does not receive any packets via this interface. It is implemented as an *IP_RAW* socket, i.e., IP packets and their corresponding header are written on the socket and thereby handed over to the bottom of the IP stack. The IP frame is not modified after being captured by the router of the sending node, except the TTL field that is decremented by one. Upon receiving the packet from the car interface, the IP stack executes a single task before handing the packet over to the data link layer that is the calculation of the header checksum.

FleetNet-unaware interface (3) Unlike the car interface, the FleetNet-unaware interface only receives packets. Both, car interface and FleetNet-unaware interface handle the FleetNet-unaware traffic with the car application system. All kinds of traffic lacking information on the layer 2.5 architecture enter the router through the FleetNet-unaware interface. The FleetNet-unaware interface is implemented as an *ipq* file handle that is provided by the *libipq* library. To obtain these packets, the file-handle has to register at the *Netlink* socket created by the *ip_queue* driver. The registration enables the router to receive packets from *iptables* via previously defined rules.

FleetNet-aware interface (4) handles the traffic of FleetNet-aware applications, i.e., applications that make use of FleetNet-specific capabilities. The interface is realized as a common UDP socket listening to port 1500. Applications send UDP packets that contain information regarding destination ID and payload. The FleetNet-aware interface is also used to send FN-aware data from other FleetNet nodes to the corresponding car application system attached to the router.

FleetNet interface (5) is the interface to the driver of the wireless network interface card. The interface is used to send and receive FleetNet packets only. A detailed description of all FleetNet packet types is given in Section 5.4. The FleetNet interface is implemented as *PF_PACKET* socket, i.e. all packets are directly handed over to the network device. The router simply assembles an Ethernet header. The IEEE 802.3 header is then transformed to the IEEE 802.11 by the logical link control. The socket is executed in promiscuous mode to enable the routing daemon to access all packets on the wireless link at MAC packet level [7].

I/O Multiplexing and Timer Management

The I/O multiplexing and timer management covers functions to handle incoming events, to control timers, and to execute procedures triggered by these events. Since for implementation design a single-threaded approach has been assumed, the *select()* function of the Linux operating system is an appropriate way to handle incoming events and thus put multiplexing into practice. This function observes a given set of file descriptors and reports changes of status to the program. Due to the fact that the router only has to multiplex between incoming packets, putting timers aside for a moment, which is basically the same as the reading on file descriptors, the `select()` function will return a particular file descriptor as soon as a read operation on that descriptor will not block. After that, the router calls the corresponding function handling the incoming packet. Unfortunately, the *libipq* does not provide the user with a file descriptor to access queued packets. Therefore, a file handle is used which does not fit into the *select()* function arguments, requiring file descriptors. Nevertheless, a way to access the packet queue via a file descriptor was found, since the file-handle contains a file descriptor which is not part of the documented API. The *select()* function also provides a timeout value, that defines the maximum waiting time for such a change of status on the file descriptors. In comparison to polling as an alternative method, the usage of timeout values in *select()* consumes less resources. The timeout argument is used to integrate timer handling into multiplexing. The router needs timers, for example, to control the periodical sending of *beacons*.

A timer is implemented as a structure consisting of four elements.

```
typedef struct sTimer { GTimer * timer;
                        guint64 fireMSec;
                        void (*callback)(void *);
                        void * argument;
                      }
```

The first element is a pointer to a timer object of the *glib* library. The second one is a 64-bit unsigned integer that stores the duration of time in milliseconds within which the timer is to be fired. The other two elements are pointers. The first one points to the callback function of the timer, i.e., the function to be called when the timer is fired. The other points to the arguments of the callback function.

All active timers are managed in a data structure efficiently implemented as a heap, i.e., an array object that can be viewed as a complete binary tree. Each node of the tree corresponds to an element of the heap storing the value of the node, in this case, the timer. The tree is completely filled and the remaining duration of the timer event is less than the remaining time of its child-nodes. Thus, the timer that is to be fired next is always at the root of the tree and can easily be extracted. The new root-timer is the smaller one of the former child-timers. The lowest key can be determined with the expense of O(1), while insertion or removing the lowest key is O(log n) [12]. Due to the usage of a heap, determining which timer is to be executed next can be realized at a very small expense.

The time that remains until the next timer expires is used as timeout argument for the *select()* function. Every loop through the *select()* function handles exactly one incoming event, which corresponds to handling one packet. This process is relatively short, and thus it is ensured that I/O handling does not block timer execution. Nevertheless, it is not guaranteed that the timer is exactly fired when it should be. Assuming that handling one incoming event takes at most a few milliseconds, this does not harm protocol performance, but one should keep in mind that the execution time of the timer is delayed to the earliest possible execution time.

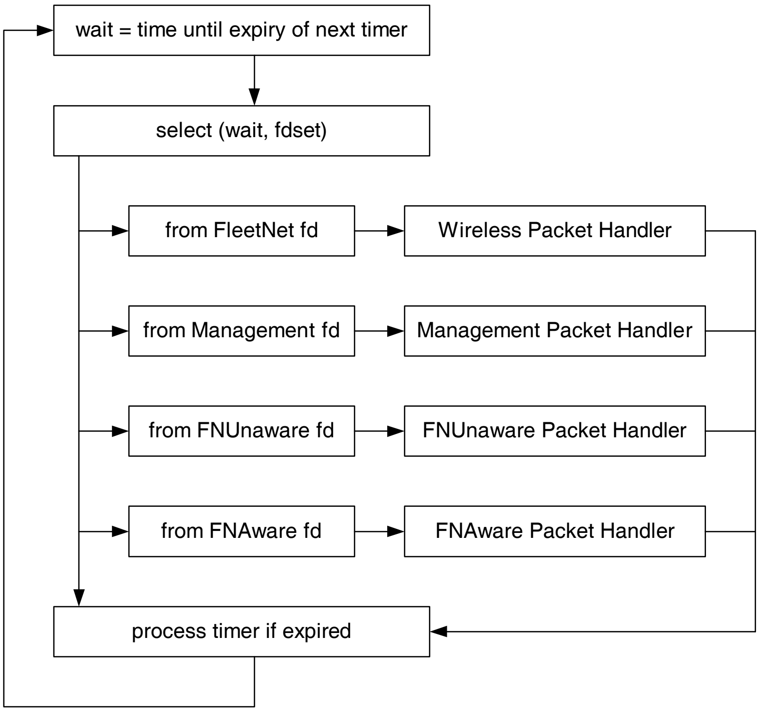


Fig. 5.5. I/O Multiplexing Architecture

The I/O multiplexer represents the core module of the routing daemon. The routing daemon executes a number of successive steps as illustrated in Figure 5.5. First, the remaining time until the next timer expires (depicted as *wait*) is determined. Second, this *wait* value is passed to the *select()* function call along with the previously initialized set of file descriptors. The program flow continues as soon as a read event occurs on one of the file descriptors or the timeout value is reached. In any case, the multiplex routine calls the appropriate handler function as shown in the figure. Since these execution steps consume a small duration of time, it has to be checked if the shortest timer expired after returning to the multiplexing routine. In that case, the timer callback function is called with the appropriate parameters. In any case, the remaining time of the next timer to be fired is determined, and the multiplexer processes the next loop.

Memory Management and Internal Data Structure

In order to ensure that the router handles every incoming event very fast, an efficient memory management is implemented to avoid multiple calls of expensive allocation and deallocation procedures at runtime, like `malloc()` and `free()` [13]. In principle, the router uses chunks of memory to store received packets or timers. When the router is started, two stacks of memory are preallocated, each consisting of ten memory blocks. One of these stacks contains blocks of 2000 bytes and is used for packets, whereas the other one contains blocks of 20 bytes which is exactly the size of the *sTimer struct* introduced in Section 5.3.1. Whenever the router needs a chunk of memory, it takes a block from the stack. The block of memory is returned to the stack as soon as it is no longer needed. In case the stack runs out of memory blocks, the size of the stack is doubled by newly allocating the same amount of blocks it already has. Since the allocation functions are modularized, the number of dynamically allocated chunks can easily be controlled.

The following part of this section will describe the main data structure used by the router, the *location table*. The location table is a single statically allocated vector with an entry for each router the node has information on. In the data structure a node is identified by the direct-hash [12] of its FLEETNET identifier. Obviously, this approach does not scale, since the location table has to contain one entry for every potential node. However, it was suitable for the number of cars used in the demonstrator. In fact, it can hardly be done more efficiently. The location table is used for neighbor information as well as for non-neighboring nodes, and also includes MAC address storage. The detailed structure of a location table entry is shown in Table 5.1.

In each entry, the location, encoded as position plus time stamp, of the corresponding router is stored. Three flags indicate whether (a) there is a pending location query (to prevent the router from starting more than one discovery cycle for the same node at one time), (b) if the node is a valid neighbor and thus may serve as next hop for the sent or forwarded packets, or (c) if the location is valid and can be used for sending packets. If a router is a valid neighbor, the location-valid flag is also set. Furthermore, each location table entry contains a field for the query number. Here, the router stores the sequence number of the last *location request* packet forwarded or answered.

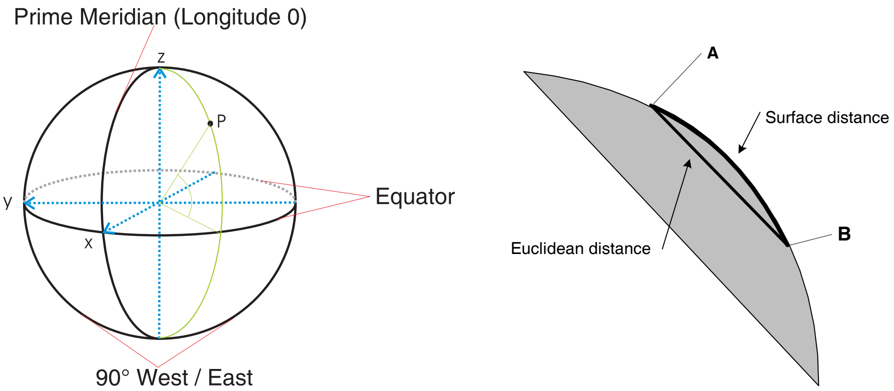
Name	Type
isNeighbor	bool
locIsValid	bool
locQueryPending	bool
MACAddress	unsigned char[]
location	LocationType
CartPosition	cPositionType
outQueue	ptr
lastLocQueryId	unsigned short
lastLocQueryTime	NetTimeType

Table 5.1. Location Table Entry

Additionally, each entry contains the MAC address of the wireless interface of the corresponding neighbor to substitute the ARP protocol. The MAC address can easily be extracted from every received packet. Lastly, there is a pointer to a queue in each of the entries. The queue is used to store packets up to a predefined number, while the *location service* tries to determine the location of the node. The direct-hash is implemented as an array of table entries (`NT`) with the mapping of `NT[ID]` of the FleetNet identifier to an entry. The maximum number of entries is a compile-time option.

5.3.2 Position Encoding

This section describes the different methods of position encoding provided by the navigation system and used in the demonstrator. The method for distance calculation is also included in this section.



(a) Geographical Coordinates (b) Euclidean vs. Surface distance

Fig. 5.6. Geography and Distances

The position provided by the navigation system of the car is encoded in well-known angle coordinates consisting of two angles, longitude (θ) and latitude (ϕ). As shown in Figure 6(a), the first defines the angle between the location and the Prime Meridian while the latter defines the angle between the location and the equator. The possible values for latitude range from -90 degrees which is the South Pole to +90 degrees, the North Pole. The maximum and minimum longitudes (+180 and -180) are on the same north south line. Both angles are floating point values.

In order to send the coordinates to the management interface of the router, the coordinates have to be transformed to integer values. Equations 5.1 and 5.2 show the conversion, which maps the maximum value of the angles to the maximum integer values providing thereby maximum possible accuracy.

$$LATI_{int32} = \left\lfloor \frac{LATI_{float}}{90} \cdot \left(2^{31} - 1\right) \right\rfloor \tag{5.1}$$

$$LONGI_{int32} = \left\lfloor \frac{LONGI_{float}}{180} \cdot \left(2^{31} - 1\right) \right\rfloor \tag{5.2}$$

To further reduce the size of the data structure for positions and therefore the routing header overhead, angles are encoded as 3-byte values, which provide a granularity of about 0.6 meters in the worst case. Regarding the accuracy of up-to-date GPS systems, this granularity is more than sufficient. Thus, angle coordinates are represented as 3-byte unsigned integers throughout the FleetNet Demonstrator.

$$LATI_{uint24} = \left\lfloor \frac{LATI_{float} + 90}{180} \cdot \left(2^{24} - 1\right) \right\rfloor \tag{5.3}$$

$$LONGI_{uint24} = \left\lfloor \frac{LONGI_{float} + 180}{360} \cdot \left(2^{24} - 1\right) \right\rfloor \tag{5.4}$$

The conversion maps the smallest possible angle to zero while the maximum possible value is mapped to $2^{24} - 1$ as shown in Equation 5.3 and 5.4. The saving of one byte per angle reduces the size of FleetNet unicast packet headers by six bytes since in each data packet the position of the source, the sender and the destination are included.

In order to calculate the surface distance a rather complex and thus expensive term of trigonometric functions is needed. A method to convert the latitude/longitude pairs into three-dimensional Cartesian coordinates is described in [13]. This method designates a point on the surface using the center of the earth as origin. The three axes are shown in Figure 6(a). The point where the Prime Meridian hits the Equator has the coordinates ($r_{earth}/0/0$). The North Pole is ($0/0/r_{earth}$). Equations 5.5 to 5.7 show the conversion of latitude/longitude pairs into Cartesian coordinates.

$$X = r \cdot \cos\left((90 - \theta) \cdot \tfrac{\pi}{180}\right) \cdot \sin\left(\phi \cdot \tfrac{\pi}{180}\right) \tag{5.5}$$

$$Y = r \cdot \sin\left((90 - \theta) \cdot \tfrac{\pi}{180}\right) \cdot \sin\left(\phi \cdot \tfrac{\pi}{180}\right) \tag{5.6}$$

$$Z = \qquad r \cdot \cos\left(\phi \cdot \tfrac{\pi}{180}\right) \tag{5.7}$$

For simplification of the conversion the Earth is approximated as a sphere with the radius $r = 6371$ kilometers. In fact, this approximation has no impact on the comparison of the calculated distances due to a specific feature of the *arc sine* function: Since this function is monotone the relations between distances stay the same under this assumption. Now, the direct Euclidean distance can be used as distance metric between two points. Obviously, the distance computed with this method is not the same as the surface distance calculated in the classical approach, since the direct Euclidean distance is measured through the earth as depicted in Figure 6(b). The square root used in the calculation of the Euclidean distance can also be omitted, as shown in Equation 5.8, since this function has also the feature to be monotonously increasing.

$$a > b \quad \Leftrightarrow \quad \sqrt{a} > \sqrt{b} \quad \forall a, b \geq 0 \tag{5.8}$$

The remaining calculation consists of three squares, three subtractions and three additions. The distance has to be calculated and compared for each neighbor of the node per forwarded packet while the coordinate conversion has to be carried out only if the location information is updated. Since the Cartesian position is only calculated when required, the computational overhead is kept low. Thus, the algorithm seems to be very suitable for efficient comparison of distance.

5.4 The Protocol

The network layer protocol used in the demonstrator offers a *best-effort datagram transport* between source and destination. The best effort delivery does not provide any guarantees in terms of packet delivery delay, sequence of packet delivery, or packet loss. Reliability is the task of upper protocol layers.

For datagram transport the protocol offers three basic services, namely *beaconing*, a *location service*, and *forwarding*. While *beaconing* is used to advertise the current location of the node to its neighbors, the *location service* provides a mean to query the geographic location of a node characterized by its identifier. *Forwarding* is the process to handle datagrams in a node and, of course, includes the sending of a datagram to other nodes.

Regarding the datagram transport, the following transport types are defined:

Unicast. Unicast is an undirected data transport service from a single node (source) to a single node (destination) by means of direct communication or by multiple hops based on specific node addresses that include node ID, position and timing information. Unicast as provided by the IP layer is directly mapped to the unicast provided by the FleetNet Demonstrator.

Topologically-Scoped Broadcast (TSB). Topologically-scoped broadcast is a data transport service from a single node (source) to all nodes in the coverage of the ad-hoc network scoped to limit the number of hops. *Single-hop broadcast* is a special case of topologically-scoped broadcast where messages are sent to the neighbors only and not forwarded for multi-hop communication. A *single-hop broadcast* is mapped to a broadcast service at the data link control layer.

Geographically-Scoped Broadcast (GSB) In contrast to Topologically-Scoped Broadcast, the addressing scope is here defined by a geometric region containing the position of the sending node.

GeoCast. GeoCast is a data transport from a single node to a group of nodes within a geographically specified region. If the destination region contains the nodes position, this forwarding mode simply maps to a geographically-scoped broadcast. If not, a more sophisticated algorithm is needed. A possible approach would be to first reach the region via unicast and then use GSB (as described in [14]).

It is important to note that the actual set of nodes reached by any forwarding mode except for unicast depends on numerous parameters like node mobility, packet size, network load, radio noise... *and* on the implementation of the protocol. In particular, the assignment of a node to the address group is evaluated when the packet reaches the node as opposed to the time when the packet was originally sent.

The remaining parts of this section outline algorithms and protocols for position-based forwarding used in the FleetNet Demonstrator. Next, location service and beaconing are described in detail (Section 5.4.1). Both services represent the building block for the forwarding service. Then, the four main types for forwarding modes are presented: *unicast packet forwarding*, *topologically-scoped broadcast*, *geometrically-scoped broadcast* and *GeoCast forwarding* (Sections 5.4.2, 5.4.3, 5.4.4, and 5.4.5).

5.4.1 Location Service and Beaconing

Position-based unicast routing requires information on

- the geographical position of the destination node, i.e., the node with which communication is desired (PD);
- the geographical position of the direct radio neighbors (PN_i).

To acquire the position of direct neighbors, a *beaconing* mechanism is used, i.e., every router periodically sends a radio broadcast containing its node identifier, which is a unique identifier for each car, and its current position as obtained from the navigation system. The time interval between *beacons* is called *beacon interval* (BINT). The corresponding frequency is called *beacon frequency* (BF). The reception of a *beacon* triggers adding/updating the correspondent entry in the router's *neighbor table (NT)*, also marking the time when the position was received. Following this mechanism, all nodes know their direct neighbors after one *beacon* round. The NT entries are soft-state [15], i.e. after a certain period of time that is greater than BI, the information on neighbors expires. This expiry interval is called *Beacon Expiry Interval* (BEXP).

The *location service* is a distributed algorithm resolving a node identifier to a geographical position. Several algorithms with different properties have been proposed in literature. For the FleetNet Demonstrator, *reactive location service* (RLS) [3], a rather simple broadcast-based approach, is used.[8] For this work, it is worth noting that any position is put in the so-called *location table* or LT. Since the location table is a superset of the *neighbor table*, the FleetNet Demonstrator provides only a single internal data structure for all location information of neighboring and non-neighboring nodes. The distinction between neighbors and non-neighbors in the *location table* is simply indicated by a boolean value. Location information can also be extracted from the FleetNet header of received data packets. Location information remains valid for an *location expiry interval* (LEXP).

5.4.2 Unicast Packet Forwarding

Forwarding is the core service provided by each node. For unicast packet forwarding with position-based routing a message carries the position of the destination node. When a node wishes to send a message to a destination, it determines the current location of the destination either by look up in its *local location table* or by using the *location service*. Using the position information as well as the knowledge of the point of time when this position was acquired, the sender selects one of the neighbor nodes as the next hop and forwards the message. In the demonstrating system the *most forward within radius* (MFR) policy is used. With the MFR policy the packet is forwarded to the neighbor with the smallest geographical distance to the destination node, thus providing the greatest progress. When the next hop node receives the message, it executes the same procedure using its own local information. If a forwarding node has a more up-to-date information of the position, the position contained in the message is adjusted. With each hop the message approaches the destination position and finally reaches the destination node.

The basic strategy to forward a message to the direct neighbor that is closest to the destination position is called *greedy position-based routing* and provided by the FleetNet Demonstrator.

[8] Please refer to the original article for a detailed description.

$$f = \arg\min_{i \in N} \{dist(PD, PN_i)\} \tag{5.9}$$

Assuming N to be the set of neighbor IDs (including the calculating node itself) and $dist(PD, PN_I)$ the distance between the packets destination node and the potential forwarder i, the best forwarder's ID is given by Equation 5.9. If this ID resolves to the nodes own ID, we are at a so-called *local optimum.* In this situation, greedy routing will not find a route to the destination, even if it would exist. For these local optima, literature proposes several *recovery strategies* [2, 16, 17]. However, these approaches are derived from graph theory and appear to work well only for static networks. Moreover, in mobile scenarios with unstable transmission ranges, severe problems can occur [18, 19]. Finding a suitable strategy for vehicular scenarios is still subject to research [20]. The default handling of a "non-greedy" packet is hence to drop the packet. Still, the router optionally queues the packet in a queue of fixed size. Whenever new information about a neighbor's position is gathered, this queue is searched for packets for which this new information may be beneficial.

The main advantage is that forwarding – given that the destination's position is known – is a purely local decision and no route setup or maintenance is required. Instead, forwarding hops are determined *on the fly*, with reasonable computation overhead scaling linearly with number of neighbor table entries. Furthermore, cars travel on the same street usually are 'greedy connected' [21].

5.4.3 Topologically-Scoped Broadcast

For topologically-scoped broadcast no position-based routing is applied. Instead the packets are flooded throughout the network limited by the number of hops HL_{init} it should travel. To prevent redundant packet processing and transmission, each packet carries a packet ID (PID) unique for each sender/packet pair[9].

The (simplified) forwarding algorithm works as depicted in Algorithm 1. The "seen packet cache" is noted as a set S_i containing the packet IDs the current node has seen for sender i. This set is initialized at node start time to be the empty set.

Algorithm 1 Topologically-Scoped Broadcast

```
packet arrives from sender $i$ with packet ID $PID$ and hop limit $HL$
if $PID \notin S_i \quad \wedge \quad HL > 0$ then
    $HL \leftarrow HL - 1$
    $S_i \leftarrow S_i \cap \{PID\}$
    rebroadcast packet
end if
```

[9] Of course, this uniqueness is only guaranteed for a certain period of time.

5.4.4 Geographically-Scoped Broadcast

The algorithm is very similar to TSB (Section 5.4.3). Instead of giving a hop limit, a geometric shape R is given, defining the region enclosing the addressed nodes. As shown in Algorithm 2, the qualifying argument for retransmitting the packet is being inside the specified region.

Algorithm 2 Geographically-Scoped Broadcast

```
local node ID is $l$
packet arrives from sender $i$ with packet ID $PID$ and Region $R$
if  $PID \notin S_i \quad \wedge \quad isInRegion(l, R)$  then
  $S_i \leftarrow S_i \cap \{PID\}$
  rebroadcast packet
end if
```

5.4.5 GeoCast Forwarding

Unlike in unicast forwarding, GeoCast forwarding is used to send packets into a geographical area. This geographical area can be approximated by predefined geometrical shapes, such as circle, rectangle, symmetric trapezium, and convex polygon [14].

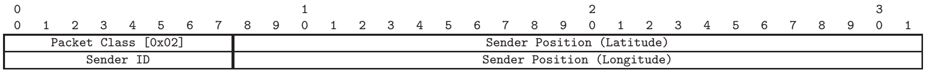
One possible solution to realize GeoCast is to regard it as the successive execution of *line forwarding* and *area forwarding*: In the first step, the packet is forwarded towards the geographical region using the greedy unicast forwarding strategy (or the recovery strategy if greedy forwarding fails). If the packet reaches a position within the geographical area, the packet is forwarded by means of the geographically-scoped broadcast.

The composition of GeoCast of *line* and *area forwarding* inherits a problem: The network topology in the addressed geographical area can be partitioned into sub-areas without network connectivity in-between.

5.4.6 Packet Types

This section introduces the different packet and header definitions implemented in the FleetNet Demonstrator. As mentioned before, *beacons* are used to acquire the positions of all nodes in radio range. RLS, which is used as location service, requires two packet types, *location requests* and *location replies*. Furthermore, to put the above mentioned traffic types into practice, basically two different data packet header types are needed, *unicast data packets* and *broadcast data packets*. The first group can be further subdivided into *FN-unaware* and *FN-aware* data packets while the latter group consists of *topologically-scoped broadcast packets* and *geographically-scoped broadcast packets*. To differ between the packet types, each packet or header begins with an 8-bit packet type identifier. During the description of the packet types, we use the terms *position* and *location* to distinguish between information that includes a time-stamp and information that does not. *Positions* are always without timestamps. Currently, the following packet types are defined in the demonstrator implementation.

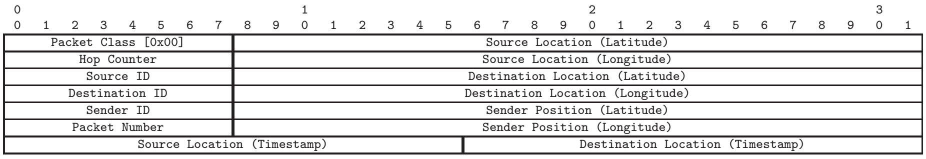
Beacons with a size of eight bytes are the smallest of all packet types, containing nothing but the FleetNet ID of the sender and its position. They are broadcast to each node in radio range in order to transmit their own positions. *Beacons* are periodically sent every *beacon interval BI*. Routers rely on received beacons to build and maintain the *neighbor table*.



0 0 1 2 3 4 5 6 7	1 2 3 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Packet Class [0x02]	Sender Position (Latitude)
Sender ID	Sender Position (Longitude)

Fig. 5.7. Format of a Beacon

FleetNet-unaware unicast data packets consist of two parts, the prefixed FleetNet routing header and the payload including IPv4 header, transport protocol header as well as the actual application payload. The header, as shown in Figure 5.8, contains an (ID,location)-pair for both, source and destination. Additionally, the position and the ID of the last hop, called sender, are included to provide piggybacked beaconing.

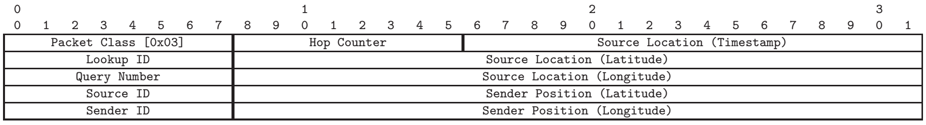


0 0 1 2 3 4 5 6 7	1 8 9 0 1 2 3 4 5	2 3 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Packet Class [0x00]	Source Location (Latitude)	
Hop Counter	Source Location (Longitude)	
Source ID	Destination Location (Latitude)	
Destination ID	Destination Location (Longitude)	
Sender ID	Sender Position (Latitude)	
Packet Number	Sender Position (Longitude)	
Source Location (Timestamp)		Destination Location (Timestamp)

Fig. 5.8. Format of a Unicast Data Packet Header

To avoid that packets can circulating forever, an 8-bit integer was added as hop counter. The packet number included in the header is unique if combined with the source ID but is not required by the routing protocol. Nevertheless, it can be used to easily trace a packet over several hops.

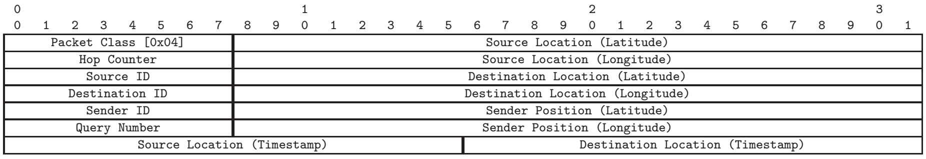
Location requests A *location request* message is used by the location service. This message is flooded to a limited number of hops defined in the header to acquire the current position of the destination. This message is also used to distribute the sender's position (piggybacked beaconing). ID and location of the source node provide an up-to-date location of the source for each node receiving the packet. To enable intermediate nodes to drop already processed packets, a query number is included in the packet.



<table>
<tr><th>0</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>1</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>2</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>3</th><th></th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th></tr>
<tr><td colspan="8">Packet Class [0x03]</td><td colspan="8">Hop Counter</td><td colspan="16">Source Location (Timestamp)</td></tr>
<tr><td colspan="8">Lookup ID</td><td colspan="24">Source Location (Latitude)</td></tr>
<tr><td colspan="8">Query Number</td><td colspan="24">Source Location (Longitude)</td></tr>
<tr><td colspan="8">Source ID</td><td colspan="24">Sender Position (Latitude)</td></tr>
<tr><td colspan="8">Sender ID</td><td colspan="24">Sender Position (Longitude)</td></tr>
</table>

Fig. 5.9. Format of a Location Request Packet

Location reply packets are issued by a node receiving a *location request* packet containing the ID of the node as lookup ID. As shown in Figure 5.10 a *location reply* packet is very similar to a data packet without payload. Again, an (ID,location)-pair is included for both, source and destination. The ID and position of the sender are required for piggybacked beaconing. It is not necessary to include the queried information as payload, since that information is already stored in the ID and location of the source.



<table>
<tr><th>0</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>1</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>2</th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th></th><th>3</th><th></th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>0</th><th>1</th></tr>
<tr><td colspan="8">Packet Class [0x04]</td><td colspan="24">Source Location (Latitude)</td></tr>
<tr><td colspan="8">Hop Counter</td><td colspan="24">Source Location (Longitude)</td></tr>
<tr><td colspan="8">Source ID</td><td colspan="24">Destination Location (Latitude)</td></tr>
<tr><td colspan="8">Destination ID</td><td colspan="24">Destination Location (Longitude)</td></tr>
<tr><td colspan="8">Sender ID</td><td colspan="24">Sender Position (Latitude)</td></tr>
<tr><td colspan="8">Query Number</td><td colspan="24">Sender Position (Longitude)</td></tr>
<tr><td colspan="16">Source Location (Timestamp)</td><td colspan="16">Destination Location (Timestamp)</td></tr>
</table>

Fig. 5.10. Format of a Location Reply Packet

Topologically-Scoped Broadcast Packet This packet type is similar to *location requests*. The only difference is, that it is not used to determine the location of a node but to transport information. Thus, the lookup ID field can be omitted and instead the application port number is included. The application port is needed to correctly send the data to the FCAN. Furthermore, the header includes a hop counter to limit the packet's reach and the ID of the sender and the source and the position of the first as well as the location of the latter. The packet number is used to prevent a router from processing the same packet twice.

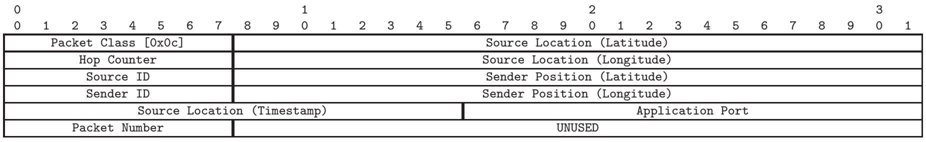


Fig. 5.11. Format of a Topologically-scoped Broadcast Packet

Table 5.2 gives an overview of the sizes of the headers of different packet classes.

Type	Size
Beacon	8 bytes
FNunaware header	28 bytes
Location Request	20 bytes
Location Reply	28 bytes
Topologically-scoped Broadcast	24 bytes

Table 5.2. Size of FLEETNET Packets Types

5.5 Experiences and Measurements

After the implementation and system integration reached a stable state, the system's performance was analyzed. The complexity of testing was steadily increased during the field trials spanning from stand-alone router systems [22] over non-moving cars [23] to a number of cars driving in a row on a circular street course [24]. The remainder of this section describes measurements made in the cars, being organized as follows: Section 5.5.1 gives static single-hop measurements allowing for a understanding of the single-hop radio performance of the communication system. This is followed by - still static - 3 hop measurements showing the bandwidth degradation with increasing hop counts. Finally, the mobile multi-hop measurements are described in Section 5.5.3.

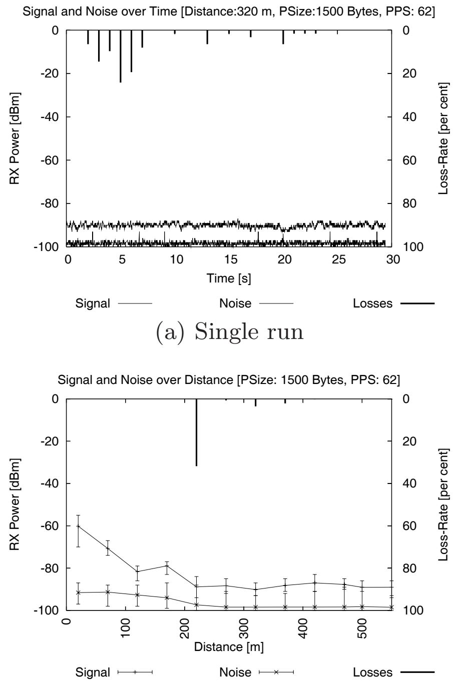
5.5.1 Static 1-hop measurements

As a first step we conducted a large number of static 1-hop measurements to determine the received power fluctuation and loss-rate over time and depending on the distance of the two cars. One car sent MAC-broadcasts of a predefined packet size (results are shown for a packet size of 1500 bytes) at a rate of 62 packets per second while the receiver stored reception power and noise gathered via functionality provided by the `iwpriv` tool [25]. Clearly, environmental factors like other cars, buildings and weather conditions have an effect on the results. Figure 12(a) shows a typical graph for measuring radio fluctuations and loss-rate for a fixed distance (320m) over 30 seconds. Note that noise and reception power are only available for successfully transmitted packets. The losses were caused by a passing non-FleetNet vehicle driving from the receiving car to the sender. The highest losses were caused as the interfering car passed the receiver (seconds 2 to 7) thereby producing a higher noise value. But even afterwards, the car caused some packet losses by reflecting or disturbing the signal. Figure 12(b) shows received power and loss-rate depending on the distance between the two cars. Even at communication ranges above 500m no noteworthy loss-rate occurred. The only problem was the unstable communication at 220m.

5.5.2 Static 3-hop scenario

For the static and mobile 3-hop measurements we decided to artificially reduce the 'transmission range' of the nodes by using a suppression mechanism to make the router drop all received packets from senders farther away than a predefined distance. By doing this the setup of multi-hop communication is facilitated, of course, at the cost of reducing spectrum re-use and by affecting corresponding MAC mechanisms. In previous tests with a laptop testbed we got aware of the fact that basic position-based routing protocols depend heavily on link stability. Upon receiving a beacon from a node, the router assumes the link to this node to be active until no beacon or packet is received from that node for a certain period of time. However, we observed frequently that beacons were received over long distances (several hundred meters) but the corresponding link quality was too low to make use of the link for successive packet transfers. Results, therefore, depend heavily on the antennae used. With standard antennae built in the WLAN card we could not manage to get results close to the one shown and discussed below in a 'real' outdoor network environment.

For the static 3-hop scenario demonstrator cars were positioned as shown in Figure 5.13 with a distance between two 'successive' nodes of approximately 150 to 200 meters. UDP as well as TCP tests were performed. To evaluate UDP performance, packets of different sizes were sent from the first node of the chain to the last. Beginning with 50 kbit/s the sending-rate was slowly increased to 500 kbit/s. The last node acknowledged packets with a small ACK packet. As shown in Figure 14(a) the maximum achievable throughput depends on the packet size since a larger number of packets results in a higher probability of collisions and thus retransmissions and even losses. A throughput of about 400kbit/s is achievable with a packet size of 1444 bytes. The graphs also demonstrate very well the well-known problem of IEEE 802.11 that the achievable rate, once it reaches its maximum, does not remain at the maximum but instead drops afterwards. Each additional packet per second degrades the performance of the network since the delay it causes is worse than the benefit of successfully transmitting the packet.



(a) Single run

(b) Accumulated runs

Fig. 5.12. Static 1-hop scenario: power and loss-rate measurements. Psize is the packet size, PPS gives the number of packets per second. The loss graph is provided by the 'superimposed' bar chart.

To evaluate TCP performance, `iperf` [26], a common network measurement tool was used. Figure 14(b) shows the TCP throughput with relation to the number of hops. As rough comparison, the results of Li et al. [27], taken by simulation and validated with a laptop testbed, are also plotted in the graph. Our values are close to Li et al.'s but a precise comparison is not possible due to the incomparable test parameters like distances of the nodes, hardware equipment or environmental influences. For 3 hops, we can achieve a TCP throughput of about 450kbit/s.

To properly analyze the effect of the quality of the various links involved, we have chosen a visualization method as depicted in Figure 5.15. Figure 5.15 shows the effects of a bad link on the network performance. The x-axis represents the different measurement points a packet and the corresponding ac-

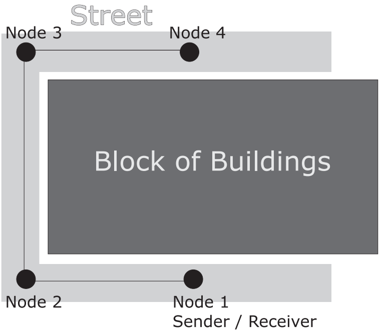
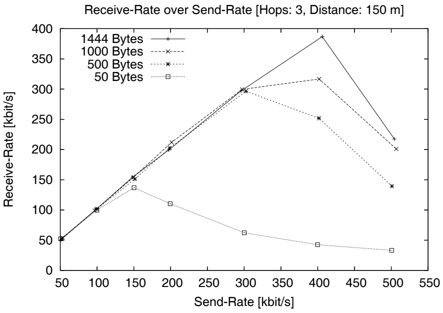
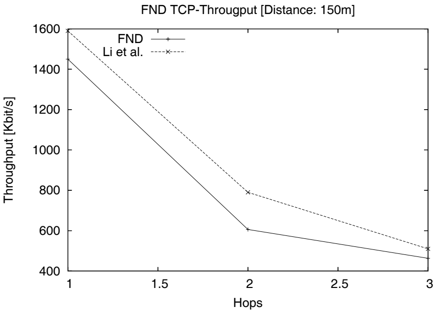


Fig. 5.13. Static 3-hop scenario



(a) UDP-throughput



(b) TCP-throughput

Fig. 5.14. Static 3-hop Scenario: throughput measurements

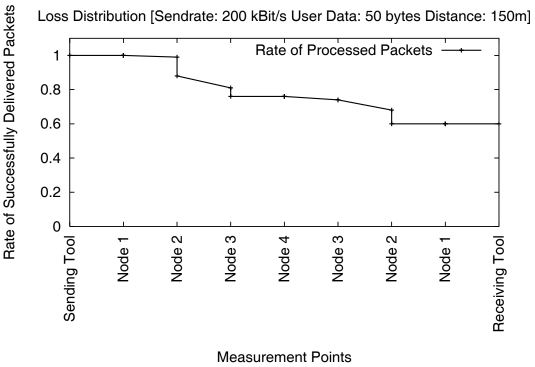


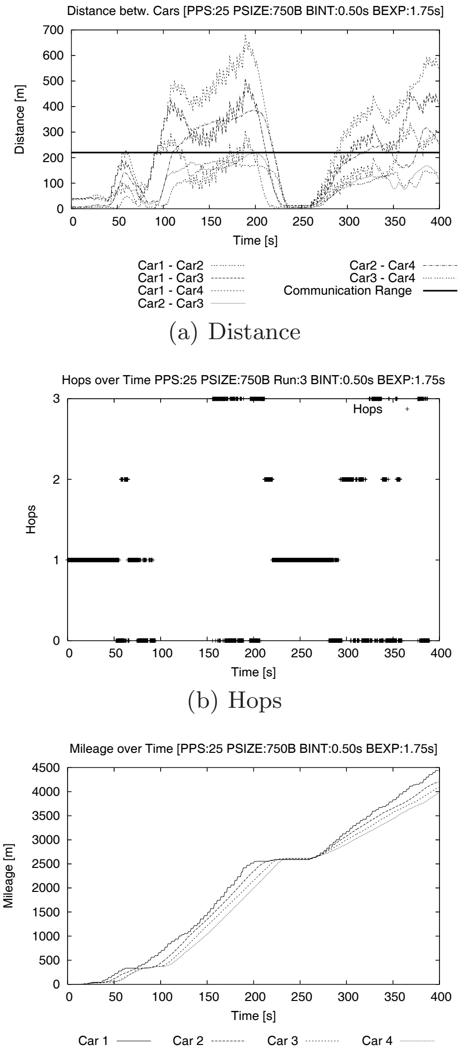
Fig. 5.15. Static 3-hop scenario: loss distribution

knowledgment have to traverse. After being created by the sending tool, the data packet has to be processed by every router. Once, a packet reaches the destination, an acknowledgment travels back to the receiving tool. There are two values per measurement point, one for the number of received packets and one for number of packets successfully handed over to the network interface. Thus, a vertical line indicates packet drops inside the router of the corresponding node due to full kernel queues. These drops are typical indications for congestion while diagonal lines, representing losses between two nodes, are link layer losses. In Figure 5.15 the bad link is between node 2 and 3, where about 10 per cent of the data packets get lost on this link while another 10 per cent are dropped in router number 2 since the kernel queues are full due to the high delay produced by a high number of 802.11 retransmission retries.

5.5.3 Mobile 3-hop scenario

Performance evaluations of mobile scenarios are a challenging task since reproducibility and comparability are much harder to achieve (if at all possible) than with fully controllable simulation environments or partly controllable static outdoor scenarios. In particular, the challenge is to keep track of all possible factors that influence results, e.g., environmental factors that influence radio propagation. In the following, we present our methodology for performance evaluations in mobile scenarios and present some key observations. However, we do not give a 'final' analysis of achievable maximal throughput or delay since the system is still under development.

In our mobile scenario, four cars were driving on a circuit that is about 5 kilometers long and depicted in Figure 5.18. The first car sent data packets of a predefined size to the last car at a rate of 25 packets per second. The last car, number 4, acknowledged the packets with small ACKs as already mentioned in the previous sections. At the beginning of the test run all cars were in each others communication range, enabling car 1 and car 4 to communicate directly. During the run we tried to built a 3-hop communication chain but had to

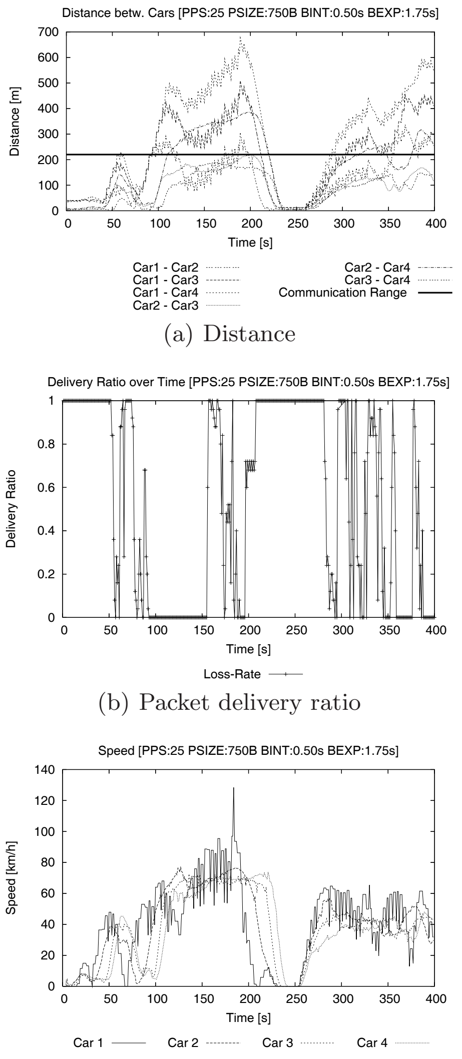


(a) Distance

(b) Hops

(c) Mileage

Fig. 5.16. Mobile 3-hop scenario: measurements



(a) Distance

(b) Packet delivery ratio

(c) Speed

Fig. 5.17. Mobile 3-hop scenario: measurements continued.

respect other cars and their right of way as well as traffic lights. Figures 16(a) to 17(c) show key parameters and measurements of a test run for a data traffic load of 150 kbit/s, i.e., the size of the packets was chosen to be 750 bytes. The first two graphs show the distance between every pair of nodes on the y-axis, while the duration of the run is given on the x-axis. For a better understanding, the maximum transmission range (suppression range of 220m) is also plotted. In other words, if a curve is above the line of the communication range, the corresponding pair of cars is unable to communicate. The distance graph is displayed twice to make it easier to analyze the other graphs in the left and right columns with respect to the distance of the cars. Figure 16(b) shows the length of the communication chain in hops. For each packet that is injected into the ad hoc network, the number of hops it had to travel to reach the destination node is displayed. A hop-count of 0 indicates that the packet has never reached the destination node. Figure 17(b) displays the packet delivery ratio on the y-axis over the duration of the test on the x-axis (aggregated in steps of 1 second). Figures 16(c) and 17(c) on the right side show the mileage of the cars as well as their speed. These two figures make is easier to reproduce the car movement and match it to the map shown in Figure 5.18.

Now, we outline the expected behavior of a vehicular ad hoc network using position-based routing and compare it to the effective behavior during the test. At the start of the run, car 1 and car 4 communicate directly. Afterwards, the distance of the cars increases. As soon as the distance is above 220 meters, the routing algorithm will react and send the packets through an intermediate hop, in this case, if possible, car 3. The fact that car 4 is no longer in communication range of car 1 is determined via a timeout mechanism. Upon receiving no position update of car 4 for at least BEXP seconds, car 1 deletes car 4 from its neighbor table and has to chose another node as forwarder. If there is no node available as next hop the packets are dropped. Looking at Figures 16(a), 16(b), and 17(b) the lossless 1-hop communication can be easily identified. It lasted until second 52, when car 4 crossed the maximum communication range to car

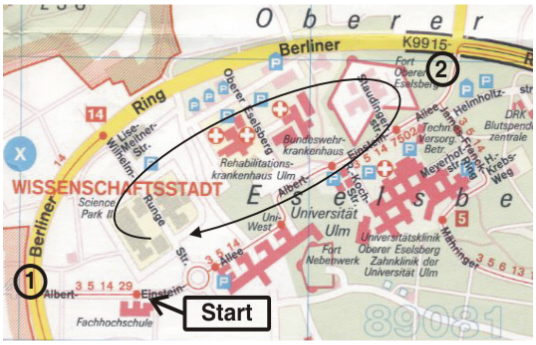


Fig. 5.18. Mobile 3-hop scenario: map of the test circuit

1. It is followed by a short period where all packets got lost, since car 1 still tried to send data directly to car 4 until it recognized after BEXP seconds that car 4 was no longer in communication range. Afterwards, car 3 forwarded the data packets as intermediate node and a 2-hop connection was established. This connection did not last long, since car 1 had to respect the right of way at point 1 on the map and car 4 came in communication range again. With car 1 having already turned right and cars 2,3 and 4 still been waiting at the crossroad, car 1 directly communicated with car 4, but the link quality was rather bad, since there were many obstacles between sender and destination. Shortly thereafter, at second 95, car 1 had no cars in communication range left and thus stopped to inject packets into the ad hoc network. At second 125, car 4 could have been reached via 3-hops, but in the meantime (LEXP was set to 20 seconds), the location information on car 4 had expired in car 1. Due to the bad link quality it took quite a long time until car 1 got the location of car 4 and was able to start to send packets over three hops.

Even in this short excerpt of the test run, some key observations can be made. First of all, the router has to have information on the quality of the link or needs a notification if a link is broken. This enables the router to react on bad or broken links without relying on timeout mechanisms. We now have implemented this 'lost link feature' to reduce packet losses after changes in the path a packet travels. Additionally, packet losses due to unstable links have to be handled by a proper transport layer, which does not exist in the current implementation of the FleetNet demonstrator. Another challenge is the location service which is based on MAC-broadcasts. Even in this small setting it took a long period of time to detect the location of car 4 over three unstable links using unacknowledged broadcast packets.

5.6 Conclusions and Outlook

In this chapter, we have presented the FleetNet demonstrator, a real-world vehicular ad-hoc network. In addition to giving an overview of the system, we have presented measurement results and deployment experiences. While the whole sub-project was built around demonstrability and to proof the concept, the future work in bringing vehicular networks on the street is manyfold. From a software engineering perspective, the software has to be brought to production quality on-board units. Since the focus of the first-to-be-deployed vehicle-to-vehicle networks will most likely be active safety rather than internet on the road', the protocol requirements are shifting from multi-hop unicast to multi-hop geocast and from end-to-end datagram routing to end-to-end information forwarding. Furthermore, new considerations concerning protocol and system architecture have to be integrated [28]. For the next steps, protocol development has to converge with real-world measurements including the precise radio hardware that will be deployed on the street. Considering this, the FleetNet demonstrator served its purpose, although there is still a way to go for a final system.

References

1. Smart networks. DaimlerChrysler Research HighTechReport (2003) 38–41
2. Karp, B.N., Kung, H.T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: Proceedings of the sixth annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '00), Boston, Massachusetts (2000) 243–254
3. Käsemann, M., Füßler, H., Hartenstein, H., Mauve, M.: A Reactive Location Service for Mobile Ad Hoc Networks. Technical Report TR-02-014, Department of Computer Science, University of Mannheim (2002)
4. Postel, J.: RFC 760: DoD standard Internet Protocol (1980)
5. ANSI/IEEE: ANSI/IEEE Std 802.11. 1999 edn. (1999)
6. ArTem OnAir. (http://www.artem.de)
7. Stevens, W.R.: UNIX Network Programming. 2nd edn. Volume 1. Prentice Hall (1998)
8. Postel, J.: RFC 768: User datagram protocol (1980)
9. Postel, J.: RFC 793: Transmission control protocol (1981)
10. Stevens, W.R.: TCP/IP Illustrated. Volume 1. Addison Wesley Longman (1994)
11. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.: RFC 1597: Address allocation for private internets (1994)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT press (2001)
13. Bentley, J.: Programming Pearls. 2nd edn. Addison-Wesley (2000)
14. Franz, W., Maihöfer, C.: Geographical Addressing and Forwarding in FleetNet. Whitepaper, available at http://www.fleetnet.de (2003)
15. Ji, P., Ge, Z., Kurose, J., Towsley, D.: A Comparison of Hard-state and Soft-state Signaling Protocols. In: Proceedings of ACM SIGCOMM 2003 Conference on Computer Communications, Karlsruhe, Germany (2003) 251–262
16. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc Wireless Networks. In: Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications (DIAL-M '99), Seattle, WS (1999) 48–55
17. Kuhn, F., Wattenhofer, R., Zollinger, A.: Worst-Case optimal and average-case efficient geometric ad-hoc routing. In: Proceedings of the fourth ACM international symposium on Mobile and ad hoc networking & computing (MobiHoc '03), Annapolis, Maryland (2003) 267–278
18. Karp, B.N.: Challenges in Geographic Routing: Sparse Networks, Obstacles, and Traffic Provisioning. Talk at DIMACS Workshop on Pervasive Networking (2001)
19. Lochert, C., Hartenstein, H., Tian, J., Füßler, H., Herrmann, D., Mauve, M.: A Routing Strategy for Vehicular Ad Hoc Networks in City Environments. In: Proc. of IEEE Intelligent Vehicles Symposium (IV2003), Columbus, OH (2003) 156–161
20. Lochert, C.: Ad Hoc Routing für die Kommunikation zwischen Fahrzeugen in Stadtszenarien. Master's thesis, Department of Mathematics and Computer Science, University of Mannheim (2003)

21. Füßler, H., Mauve, M., Hartenstein, H., Käsemann, M., Vollmer, D.: A Comparison of Routing Strategies for Vehicular Ad Hoc Networks. Technical Report TR-02-003, Department of Computer Science, University of Mannheim (2002)
22. Möske, M.: Real-World Evaluation of a Vehicular Ad Hoc Network using Position-Based Routing. Master's thesis, Department of Computer Science, University of Mannheim (2003)
23. Hartenstein, H., Füßler, H., Mauve, M., Franz, W.: Simulation Results and Proof-of-Concept Implementation of the FleetNet Position-Based Router. In: Proceedings of the IFIP-TC6 8th International Conference on Personal Wireless Communications (PWC '03), Venice, Italy (2003) 192–197
24. Möske, M., Füßler, H., Hartenstein, H., Franz, W.: Performance Measurements of a Vehicular Ad Hoc Network. In: Proceedings of the IEEE Vehicular Technology Conference (VTC'04 Spring), Milan, Italy (2004)
25. Wireless Tools for Linux. (http://www.hpl.hp.com/personal/ Jean_Tourrilhes/Linux/Tools.html)
26. IPerf. (http://dast.nlanr.net/Projects/Iperf/)
27. Li, J., Blake, C., DeCouto, D.S.J., Lee, H.I., Morris, R.: Capacity of Ad Hoc Wireless Networks. In: Proceedings of the seventh annual ACM/IEEE International Conference on Mobile computing and networking (MobiCom '01), Rome, Italy (2001) 61–69
28. Füßler, H., Torrent-Moreno, M., Transier, M., Festag, A., Hartenstein, H.: Thoughts on a Protocol Architecture for Vehicular Ad-Hoc Networks. In: 2nd International Workshop on Intelligent Transportation, Hamburg, Germany (2005) 41–45

6

Internet Integration

Bernd Bochow[1] and Marc Bechler[2]

[1] Fraunhofer Institute for Open Communication Systems, Berlin, Germany
Bernd.Bochow@fokus.fraunhofer.de
[2] Technical University of Braunschweig, Braunschweig, Germany
bechler@ibr.cs.tu-bs.de

Summary. A vehicle equipped with VANET technology can be regarded as a mobile sub-net rather than a single roaming node. Devices that are part of the vehicle may request services from stationary servers in the Internet or from other vehicles. At the same time they can host and export services to both other vehicles and to Internet-based clients. Vehicles in a VANET will therefore become an integral part of the Internet. This chapter describes the FleetNet approach of integrating a Vehicular Ad-hoc Network (vanet) into the global Internet. Functional requirements, system architecture and implementation constraints are described in detail. Additionally, the FleetNet Internet Integration demonstrator is presented, consisting of a limited number of vehicular and fixed nodes. Concluding this chapter, a summary of lessons learned from this trial platform and an outlook regarding the future role of Internet Integration in the context of vanets is given.

6.1 Introduction

Mobility in communications nowadays almost always refers to personal communications. Location-based, personalized and context-aware applications are state of the art in 3$^{\text{rd}}$ Generation (3G) cellular networking and are now broadly accepted. But vehicular communications not only can be seen as a special case of personal communications characterized by high velocity mobile clients. It also shares all facets of ubiquity in modern communications: passengers to vehicle, vehicle to vehicle, passengers via vehicle and vehicle to infrastructure communications. In the form of human-machine and machine-to-machine communications, this relies on location-awareness and context-awareness regarding passengers, vehicle and groups of vehicles.

Mobile office, infotainment applications and traffic telematics based on cellular communications, broadcasting services and hybrid communications are state of the art in vehicular communications. It has also been shown that traffic safety can benefit significantly from vehicular communications (e.g., [1]). The latter is likely to be 'invisible' to the user under normal conditions. But, since users will expect to have a noticeable benefit from a newly introduced communications system, providing Internet access capabilities at stationary FleetNet access points or even while on the road will be a crucial point for market introduction.

Information appliances in the car (providing information about weather or road conditions ahead, tourist or marketing information, etc.) will access resources via the Internet e.g. to assist route planning. Furthermore, vehicles will act as sensors providing information (e.g., local traffic or weather, obstacles in sight, free parking lots) to other vehicles or clients located somewhere in the Internet.

A network of vehicles clearly benefits from its capability to access the Internet. Vehicles not only act as mobile clients but also might take the role of a server. For example, a VANET-enabled vehicle might provide information collected from its surroundings via multi-hop car-to-car communication as a service to a client situated in the Internet. This client in turn might collect information from several sources to provide e.g. real-time traffic information services to its subscribers. Basically, such a service will not need any server-side storage of traffic data such as existing Floating Car Data (FCD) based solutions, but will rely on a distributed live database located in the VANET. Additionally, using geographical addressing within the VANET to obtain information from a certain location rather than a dedicated vehicle, enables masquerading of individual vehicles for this kind of anonymous information services.

Hence, a VANET as described in this section creates an inter-vehicle communications platform as an integral part of the Internet. It provides network mobility rather than personal mobility and supports service provisioning as well as service subscription. Thus, the term *Internet Integration* has been chosen here to describe the role of a VANET as an 'Internet on the Road'.

6.2 Internet Access Technologies for Mobile Ad-hoc Networks

6.2.1 Challenges in providing Internet Connectivity for Mobile Ad-hoc Networks

According to Murphy et al. [2], an ad-hoc network is "*... a transitory association of mobile nodes which do not depend upon any fixed support infrastructure. [...] Connection and disconnection is controlled by the distance among nodes and by willingness to collaborate in the formation of cohesive, albeit transitory community ...*". In fact, providing Internet access capabilities to a mobile ad-hoc network demands for some type of infrastructure, i.e., a wired or wireless core network and gateway functions or access routers for interfacing this network situated at logical network boundaries. If we assume that gateways are dedicated special function service nodes, the number of available gateway nodes likely will be much less than the number of nodes that do not provide such type of service. Thus, gateway functions can be seen as a scarce network resource, and the network will tend to make use of gateways, i.e., to maintain connectivity to the fixed Internet as long as possible, regarding efficiency (in terms of network resource allocation) rather than geographical node distance.

To construct a gateway 'infrastructure' in the context of an 'infrastructureless' mobile ad-hoc network rises a number of demanding challenges:

- How to achieve an interworking between potentially incompatible domains of mobility support regarding end-system and network mobility?
- How to cope with a frequent loss of connectivity due to a (geographically) scattered topology caused by, e.g., sparse gateway density, low gateway penetration or frequent network re-organization?
- How to locate a gateway service in a self-organizing topology, i.e., how to organize service announcement and detection and to optimize service announcement propagation in an ad-hoc network?
- How to organize gateway association? That is, how to optimize the service area of a single gateway with respect to client density?
- How to cope with the presence of multiple overlapping gateway service areas, i.e., how to compute, select and maintain the optimal association with one out of many gateways?
- In case of multi-hop communications, how do we determine the optimal route to a gateway with respect to network resource utilization and connection lifetime?

VANETs mainly are characterized by frequent reconfigurations of network topology and a constantly varying number of nodes to participate in an ad-hoc network. This is due to a tremendous node mobility in conjunction with a limited one-hop radio communications range and a potentially large number of nodes wide spread over a geographical area – possibly moving each into different directions. This bears on special requirements for Internet Integration such as

- scalability of components, protocols and services;
- fast service detection, registration and de-registration;
- robustness against transmission delay variation and packet loss;
- end-to-end encryption; and
- intelligent resource control.

Regarding the enormous number of vehicles that potentially constitute the VANET, scalability of Internet Integration is an issue. Inter-vehicle communications in FleetNet gains from spatial diversity and locality of information – but any Internet traffic must cross a single gateway nearby.

Thus, gateways and other required back-end components need to be scalable either in number (spatial partitioning) or in performance (functional partitioning) depending on the intended geographical service area, i.e., the expected number of vehicles to serve. Additionally, gateway association –

that is, gateway detection, connection set-up, tear-down, handover and Mobile Node (MN) registration and de-registration – must be fast and efficient in terms of protocol overhead, time delay and bandwidth requirements in order to continuously maintain connectivity as long as feasible. Since service announcement procedures in a FleetNet VANET mostly rely on geographical broadcast messaging, message size and bandwidth consumption are an issue too.

Currently, Quality of Service (QoS) is not a major issue in FleetNet Internet Integration. In multi-hop packet forwarding, packet loss probability and packet delay variation depends on the total hop count between gateway and MN. For a growing communication distance, the hop count required to bridge the distance increases equally and in turn leads to a variation of the total number of hops needed, due to the underlying stateless hop-by-hop packet forwarding. Nevertheless, resource control within the FleetNet VANET is mandatory due to the fact that any kind of traffic bound to the Internet must not degrade the QoS in local high priority or delay-sensitive inter-vehicular communications that might bear safety related data, e.g., vehicle-to-vehicle emergency messages.

In general, security in ad-hoc networking is an active research topic and is of utmost importance in vehicular applications. In particular, multi-hop communications implies that since every node might be a forwarding or routing node, the nodes are theoretically able to monitor or tamper end-to-end traffic (malicious nodes). Hence, providing Internet connectivity to a VANET strictly demands for encryption of user data. Since communications between a MN and a gateway is most sensitive against tampering, this leg must be protected even if end-to-end communications is unprotected, such as in simple Web browsing.

6.2.2 Mobility Support in Wireless Access Networks

Lower layer mobility is provided inherently by the FleetNet geographical ad-hoc routing described earlier (ref. section on Position-Based Routing for Car-to-Car Communication). This has been optimized for car-to-car communications in a VANET. But providing IP-mobility and Internet access on top of VANET-mobility is of utmost importance for a future commercial roll-out. Thus, we need to look at existing options for IP-mobility support in wireless networks first in order to provide a foundation for mobility support in FleetNet Internet Integration.

Wireless Internet

According to Banerjee et al. [3], mobility support in the wireless Internet – with respect to mobility protocols – can be categorized as *Application Layer Mobility* (e.g. the Session Initiation Protocol (SIP)), *Transport Layer Mobility* (e.g., TCP-Migrate), *Network Layer Mobility* (e.g., Mobile IP), and *Subnetwork Mobility* (e.g., the General Packet Radio Service (GPRS)).

Subnetwork mobility is provided by link layer mechanisms solely. It applies to 'invisibly' changing a MN's layer 2 point of attachment within a single subnetwork.

Network layer mobility is intended to provide network-level transparency, that is, to hide changes in a MN's network-level point of attachment. Higher layer protocols thus do not need to cope with a change of, e.g., the IP address of a MN due to roaming between several subnetworks. In this context, the term 'micromobility' often refers to MN mobility within a single Internet domain, while 'macromobility' describes MN mobility across several Internet domains.

Since a major design decision in the FleetNet context was to leave IP-based (FleetNet-non-aware) applications and protocols unharmed, application and transport layer mobility approaches are of less importance here due their impact on application and transport protocol interoperability.

GPRS

Currently, 3G cellular networks such as the Universal Mobile Telecommunication System (UMTS), respectively the GPRS, are the sole large scale access networks providing end-system mobility for voice and data communications. In GPRS – the packet switching domain of a UMTS network – a MN needs to initiate a routing area update procedure when roaming from one Routing Area (RA) to the next as depicted in fig. 6.1. Simplifying, the MN detects that a new cell has been entered by comparing the cell's identity with the one stored in the MN's Mobility Management (MM) context. The MN detects that a new RA has been entered by periodically comparing the RA identity stored in its MM context with that received from the new cell. When the MN camps on a new cell, either a *cell update* or a *routing area update* is required. In this process, the portion of the MM context that is stored in the Home Location Register (HLR) is updated with the MN's new location. In the UMTS, RAs and Location Areas (LAs) are independent of each other. That is, a RA does not necessarily need to consist of a number of contiguous cells that cover a coherent geographical area.

Wireless LANs

In the variety of present Wireless local area network (WLAN) infrastructures, Access Points (APs) mainly act as layer 2 bridges. Mobility support is based on a communication between APs via a Layer 2 distribution network (cf. fig. 6.2). Inter Access Point Protocols (IAPPs) (e.g., [4]) are vendor specific and commonly rely on a set of proprietary layer 2 and IP-based protocols for AP announcement, AP management and the distribution of bridge table updates and authentication data. An IAPP hand-over protocol is responsible for implementing smooth hand-over of roaming MNs. A vendor-independent IAPP has been specified by IEEE 802.11f [5], which allows roaming between

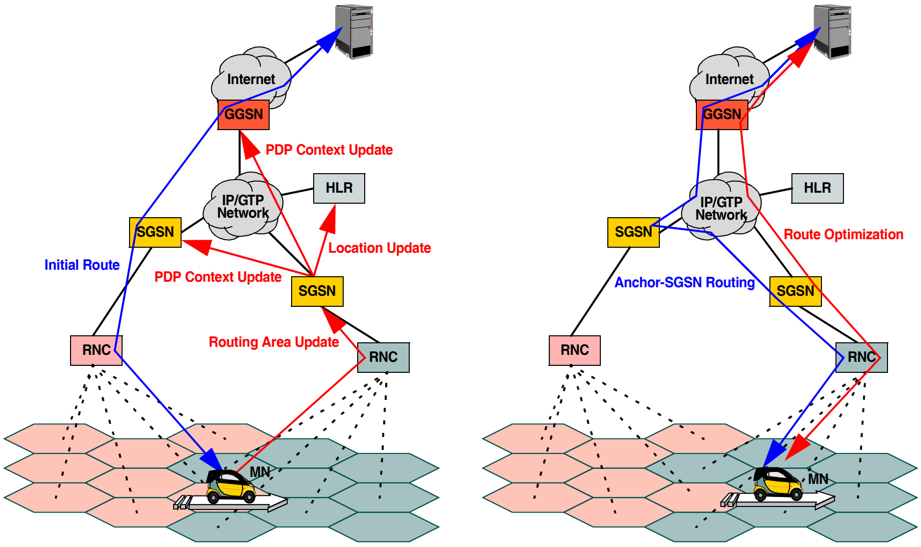


Fig. 6.1. GPRS Mobility in 3G Cellular Networks

multi-vendor APs and provides fast hand-off procedures, but is not yet widely accepted.

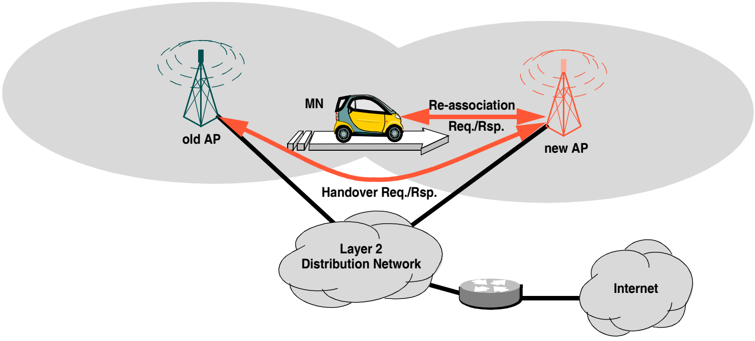


Fig. 6.2. Mobility in common WLAN Architectures

Current activities at the 3GPP and the IEEE target at the integration of WLAN architectures into 3G cellular networks (3GPP: [6]) or public networks in general (IEEE P802.11 WIG: [7, 8]) as a complementary technology. Although approaches differ, both groups currently focus on the specification of interfaces and interworking functions such as Call Control (CC) and Authentication, Authorization, Accounting (AAA). MM support is not yet fully specified.

3GPP concentrates on the use of hotspot WLANs to enhance the performance of 3G cellular networks. Macromobility will be provided by the 3G MM core components.

IEEE P802.11 WIG rather aims on an architecture to support interworking with arbitrary networks. This is likely to evolve public hotspot architectures based on RADIUS [9] and VPN that currently do not support macromobility and AAA interworking.

Mobile IPv4

Mobile IPv4 [10] provides protocol enhancements for a transparent routing of IP packets to MNs in the Internet. A MN is always identified by its home address, regardless of its current point of attachment to the Internet. The home address is located in the home network of the MN. While situated away from its home network, a MN is also associated with a Care-of Address (CoA), which provides information about the MN's current point of attachment to the Internet. Therefore, Mobile IPv4 deploys an agent-based system, which comprises a Home Agent (HA) and a Foreign Agent (FA) as illustrated in fig. 6.3. If a Correspondent Node (CN) in the Internet sends IP packets to the home address of the MN, packets will be routed to the home network of the MN. If the MN is currently located in a foreign network, the HA accepts the IP packets on behalf of the MN; it then encapsulates the IP packets and tunnels them to the current CoA of the MN. Hence, the packets are routed through the Internet to the FA, which decapsulates the IP packets and forwards them to the MN. Vice versa, IP packets from the MN to the CN are transmitted either directly to the CN, or they are first tunneled back to the HA ('reverse tunneling'), which unpacks the IP packets and routes them through the Internet to the CN.

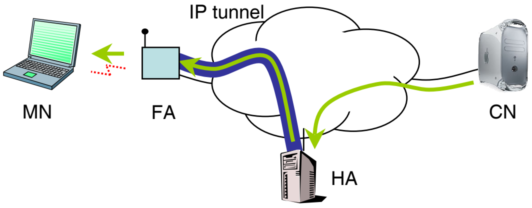


Fig. 6.3. Mobile IPv4

Mobile IPv4 basically requires that a MN is able to discover a FA when it enters a foreign network. For this, a FA advertises its presence (and its CoA) periodically using *agent advertisements*. Additionally, the transmission of agent advertisements is triggered by the MN; it therefore multicasts an *agent solicitation*, which will be responded by the FA with an agent advertisement. If the MN receives agent advertisements of a new FA, it has to register itself with both the FA and the HA in order to notify the HA about its new location.

An alternative deployment of Mobile IP is based on co-located CoAs, which shifts the FA functionality into the MNs. If a MN enters a foreign network, it uses a temporary co-located CoA from within the foreign network. The co-located CoA can be configured either statically by the user or dynamically using, e.g., the Dynamic Host Configuration Protocol (DHCP) [11]. The MN then registers its co-located CoA with its HA. Thus, the MN has a topologically correct IP address which can be used for further communication; an explicit FA is not necessary. Communication is similar as described above, except that the IP packets are tunneled from the HA directly to the MN.

Mobile IPv6

Mobile IPv6 [12] manages roaming of MNs between wireless IPv6 networks. For this purpose, Mobile IPv6 is based on an agent-based system similar to Mobile IPv4 using co-located CoAs. This way, Mobile IPv6 does not use any FAs. If a MN moves to a new point of attachment in another subnet, it has to acquire a new valid CoA from within the foreign network. When the MN is in a foreign network, its HA in the home network acts as a representative of the MN. In contrast to Mobile IPv4, the MN in Mobile IPv6 has to register its current CoA with its HA and with the CNs it currently communicates with. The association made between the home address and the current CoA of a MN is also called a binding.

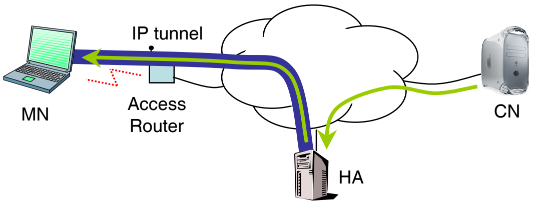


Fig. 6.4. Mobile IPv6

A MN detects that it has entered a new subnet by analyzing router advertisements periodically sent by the access router of the foreign network. The MN also can request the access router to send a router advertisement by multicasting a router solicitation [13]. Based on the information obtained, the MN then creates a valid IPv6 address as described in [14]. Finally, the MN updates the bindings in the HA and its current CNs by sending binding updates to these nodes.

In order to communicate with a MN, a CN first queries its stored bindings for the MN's address. If it finds an (updated) binding for the MN, it will communicate with the MN directly using its current CoA. Otherwise, the CN sends the IPv6 packets to the home address of the MN. The HA then receives the IPv6 packets and tunnels them to the MN's current CoA as illustrated in fig. 6.4. The other direction from the MN to the CN follows the same way, i.e. IPv6 packets from the MN are tunneled back to the HA, which decapsulates the IPv6 packets and forwards them to the CN.

Mobile IPv6 bears on various modifications of the original IPv6 protocol to implement its functionality: new IPv6 headers, new protocol extensions, new ICMPv6 messages, and a modified neighbor discovery algorithm [12]. Additionally, enhancements for the handoff procedure, such as the simultaneous support of several access routers or a forwarding mechanism for former access routers are provided. Moreover, further extensions were proposed to improve the handoff procedure, such as Hierarchical Mobile IPv6 [15] or Fast Handoffs [16].

6.3 FleetNet Approach of Internet Integration

6.3.1 Functional Requirements to the Communication Platform

The FleetNet VANET forms a communication platform intended to support both inter-vehicle and Internet communications equally. This leads to a wide range of requirements regarding reliability and availability of communication services in order to cover highly demanding driver assistance and safety-related functions as well as IP-based user communications for car-to-car and Internet-based applications. The main application area of FleetNet Internet Integration is in providing network services for IP-based user communications. Although driver assistance and distributed FCD may gain from Internet access, requirements are assumed to be too strict to rely on the uncertainties of a slowly evolving architecture. Nevertheless, Internet Integration can take the role of a market enabler for FleetNet VANET deployment.

Gateway Infrastructure

From possible market introduction scenarios a number of major requirements for the FleetNet Internet Integration architecture evolve:

- The gateway architecture must support standard IP protocols in end-to-end communications as well as any wide-spread Internet-based application considering technical (e.g., in streaming media transport and QoS support in a mobile ad-hoc network) or regulative (e.g., in providing broadcasting services) restrictions.
- The gateway infrastructure must allow to be set-up without any centralized organization or topological planning, but must not block out options that require parts of the resulting infrastructure to be established and maintained by some commercial provider. Additionally, the infrastructure must be able to operate without any centralized network management, but must allow configuration and management of a topological or organizational subset by some service provider.

- The service architecture must be optimized for peer-to-peer services to foster the unmanaged evolution of a gateway infrastructure, but must allow for centralized and managed services as well.
- Infrastructure components must be cost-effective and should consist of off-the-shelf hardware and software as much as feasible to promote deployment based on peer-to-peer scenarios.
- Gateways must be able to provide multiple services – Internet access should be one out of many services supported – and must allow sharing of infrastructure components by multiple service providers.

Clearly, the unmanaged evolution of a gateway infrastructure will not allow for any overall network planning. Availability of Internet access via gateways in terms of geographical coverage and available bandwidth cannot be foreseen easily. Nevertheless, estimates can be obtained from similarities to the upcoming WLAN hot-spot infrastructure in conjunction with FleetNet VANET multi-hop IP packet forwarding capabilities, virtually extending the service coverage of a single gateway far beyond its radio coverage (cf. fig. 6.5).

The geographical service area of a FleetNet Gateway (FGW) therefore is basically limited only by distance, i.e., by the soft degradation of link reliability due to the increasing number of hops required to reach the FGW. Thus, FGWs need to limit their service area actively and MNs need to decide on the best selection out of all available FGWs.

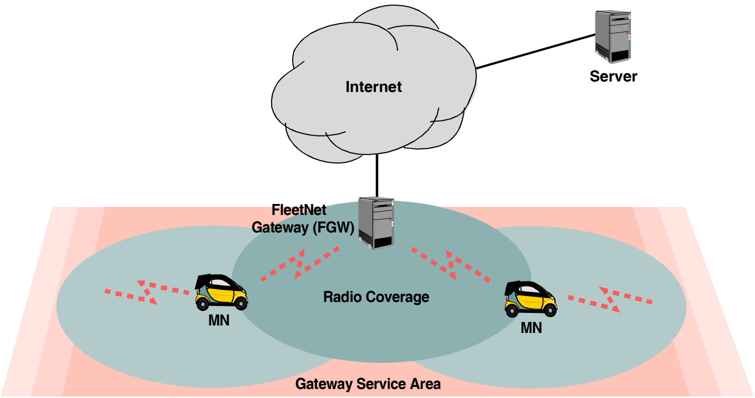


Fig. 6.5. Gateway service area in the FleetNet Internet Integration architecture

To overcome limitations due to an initially sparse FGW density, measures can be taken to indirectly foster the evolution of a gateway infrastructure, e.g.:

- Provide *deployment applications*. That is, applications initially preferred for user communications should be designed for robustness against occasional disruption of Internet connectivity – which makes location-based messaging and store-and-forward applications to strong candidates rather than interactive applications (ref. chapter on ‘The Pinboard Application - Location-Based Messaging for Vehicular Communications’).

- Achieve interoperability with WLAN hot-spots. That is, FLEETNET Internet Integration should allow to make use of WLAN hot-spots and should allow to provide hot-spot services as well – which means that vehicles can access WLAN hot-spots (which actually is not a service of the FLEETNET, but might rely on a shared radio hardware). Furthermore, FGWs can provide hot-spot access e.g. as a third-party service.

In the FLEETNET VANET each node is a routing node that is capable of forwarding packets toward their geographical destination. The same must hold for gateways: from an architectural point of view gateways are fully functional but immobile routing nodes providing services to other VANET nodes. This opens up an additional view on the role of gateways within a VANET:

- FGWs actively contribute to initially establishing a mobile ad-hoc network.
- FGWs can be used to ‘tunnel’ VANET data packets through the Internet closer toward their geographical destination via a distant FGW.
- Any MN capable of connecting to the Internet, e.g., via a cellular network, can be used to provide Internet access to other MNs.

For the time being, the latter is only of academic interest. Although a mobile FGW maybe technically feasible, it throws up questions about billing the service and restricting the cellular access to certain MNs, as well as the general usefulness of a network based access service compared to a per MN cellular access. Nevertheless this option could be of some significance for organizational use such as for rescue or maintenance services, if a mobile FGW could be placed in an optimized position.

Mobile Node Architecture

One of the key components of FLEETNET Internet Integration is in the capabilities of the communication platform to support IP-based services provided by a MN. Clearly, a vehicle can act as a client to an IP-based server either situated in the VANET or somewhere in the global Internet. But vehicles also can operate a co-located server, such as an embedded Web server, e.g., as an information appliance to neighboring vehicles or clients situated somewhere in the global Internet. This implies the assignment of globally unique IP addresses to cars, and, since a vehicle forms a mobile subnet, the assignment of a dedicated address space to each car in order to unambiguously reference distinct on-board elements.

The address space assigned to each car is based on the prefix scheme depicted in fig. 6.6. It relies on IPv6 addresses merged with the FLEETNET Node Id (FNID) which constitutes the globally unique layer 2 address of a FLEETNET node. Hence, a FLEETNET vehicle forms a mobile IPv6 subnet.

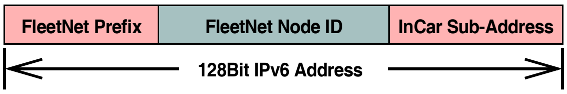


Fig. 6.6. FleetNet IP Address Scheme

In an architectural view, a FleetNet MN consists of a router element providing geographical packet forwarding functions for neighboring nodes as well as providing access to the VANET for clients and servers attached to the vehicle on-board network infrastructure (cf. fig. 6.7). IP-mobility support is to be provided by the FGW infrastructure. Additionally, support for client mobility – e.g., passenger's notebooks, that connect to the on-board network and need to gain access to the Internet via FleetNet Internet Integration – has to be provided by the MN's service architecture. In detail, this demands for Mobile IPv4/IPv6 over FleetNet VANET support.

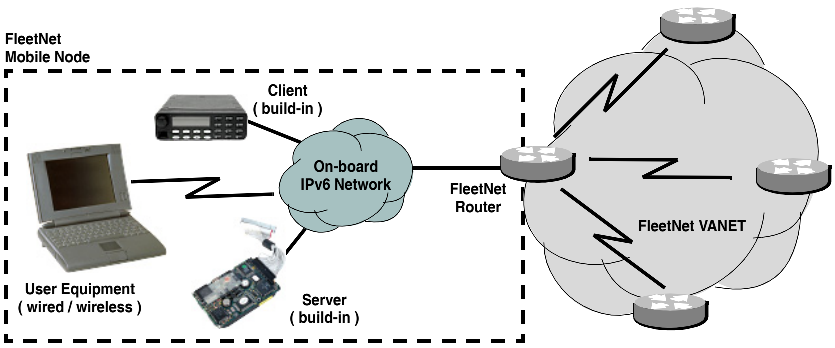


Fig. 6.7. FleetNet VANET MN On-board Network architecture

Mobility Management

From the discussions done so far, it becomes clear that IP-mobility support in the FleetNet VANET cannot be achieved by simply adopting one of the well known options discussed earlier (cf. sect. 6.2.2). But, design decisions taken finally make use of elements found in all of these to satisfy the demands of FleetNet Internet Integration.

Inter-gateway communication similar to an Inter Access Point Protocol requires FGWs to communicate via a layer 2 distribution network and to act as layer 2 bridges in order to route IP packets originating in the Internet toward their destination MN. With the exception of clustering FGWs into a LAN infrastructure, this is neither scalable nor feasible since FGWs are distributed geographically and might connect to the Internet by any available means. IP-mobility support for this type of architecture thus would require Mobile-IP in any way to provide macromobility.

In the FleetNet VANET no assumptions can be made about gateway topology – including geographical information, such as FGW location and serviced area. Additionally, a gateway service very well could be restricted, e.g., to MNs that satisfy certain conditions (a certain manufacturer, owner, car rental company, or similar). Hence, a FGW usually cannot make any assumption about neighboring gateways and thus cannot establish communications to another FGW without support by some central element keeping track of the status, network address and location of a multitude of FGWs.

Such an element has been defined by the FleetNet Proxy (cf. sect. 6.3.2) for multiple reasons. Clearly, gateways could take the role of a proxy themselves. But this would lead to an increase in gateway complexity, would impose problems regarding scalability, and would require FGWs to announce their presence to other FGWs using broadcast or multicast-based communications.

In order to extend the geographic VANET coverage under sparse MN density conditions, the FleetNet VANET makes use of direct gateway to gateway communications (cf. sect. 6.3.5). In this context the FleetNet Proxy provides information such as location and IP-address about associated gateways in order to establish a mesh of gateways via point-to-point communications.

Gateway selection in the FleetNet VANET must be initiated by MNs due to the lack of a centralized Mobility Management. That is, FGWs need to distribute service advertisements and MNs need to select the FGW best suited to their needs. For this, a FGW needs to announce its IP-Address, location, type of service, and, optionally, its geographical area of service as well as status information to assist MNs in FGW selection. In this, the service area of a FGW resembles the Routing Area in the GPRS.

Using multi-hop communications, FGW advertisements may travel much farther than the serviced area limits, which allows MNs to learn about FGWs well before approaching their service area. Nevertheless, to avoid flooding the VANET with service advertisements of multiple FGWs, FleetNet geographical broadcast must be used to limit the impact on network bandwidth.

Macromobility in FleetNet Internet Integration requires IP mobility support for IPv6. Mobile IPv6 was designed for the mobility support of MNs moving between wireless IPv6 networks. Applied to the FleetNet scenario, vehicles act as MNs, FGWs take the role of access routers, and HAs are located in the home networks of the vehicles. However, Mobile IPv6 cannot be used to integrate multi-hop ad-hoc networks into the Internet. The most fundamental problem is that most protocol mechanisms in Mobile IPv6 require link-local multicast support. For example, router solicitations and router advertisements for identifying the access routers are sent to link-local multi-cast addresses. This feature is harmful in a VANET in several ways:

- The use of link-local addresses implies that router advertisements only are transmitted to MNs within the radio range of a FGW.
- In the FleetNet a vehicle typically acts as both a router and an end system. This way, a vehicle will permanently receive router advertisements from its adjacent vehicles. This implies that all neighboring vehicles are assumed to be access routers to the Internet, which is not valid for the FleetNet VANET.
- CoAs impress a hierarchical address structure for the VANET that can be hardly established and maintained due to the mobility of vehicles.
- Multicast support in the FleetNet VANET will be available in the form of a geographical multicast only. A multicast address scheme will not be available – which makes it a demanding task to map router solicitations and router advertisements onto a VANET geographical multicast transport.

Furthermore, protocol mechanisms of Mobile IPv6 are neither scalable nor efficient for communication in a VANET. Vehicles permanently have to update their link-local addresses and they permanently have to find new FGWs by transmitting router solicitations while on the move. Due to the large number of vehicles in a VANET, this causes significant overhead even when FGWs are not in range.

6.3.2 Architectural Overview

The key principle in FleetNet Internet Integration is in the combination of a proxy-based communication architecture with network layer mobility support. This approach meets the following requirements:

- IP mobility support and Internet access for vehicles.
- Integration of lower layer (ad-hoc-network) mobility.
- Transport layer efficiency.
- Scalability.

The basic FleetNet scenario for Internet Integration is illustrated in fig. 6.8. If a client or server situated in the on-board network of a MN directs an IP packet toward a CN situated somewhere in the Internet, FleetNet Internet Integration components co-located to the FleetNet router detect an IP address not associated with a VANET node. Thus, the packet is forwarded to a FGW crossing the ad-hoc network until it reaches the selected FGW. In this, the selected FGW acts as the MN's default router. The packet then continues its way via the FGW network toward an associated FleetNet Proxy. Here, the FleetNet Proxy acts as an access router to the Internet and forwards the packet toward the CN.

An IP packet originating at the CN will reach the FleetNet Proxy associated with the MN via the Internet. The FleetNet Proxy then acts as an access router and forwards the packet toward the FGW currently associated

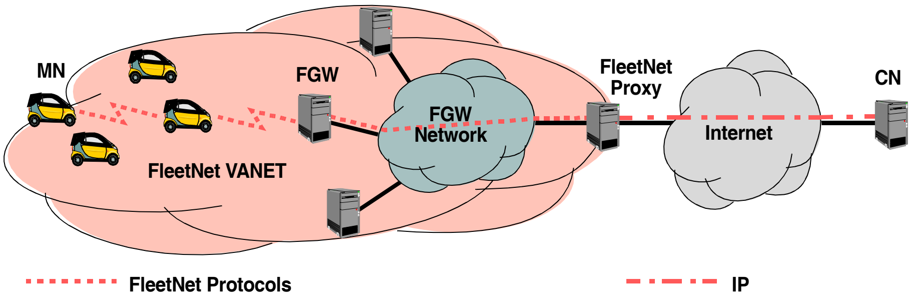


Fig. 6.8. Basic FLEETNET scenario for Internet Integration

with the MN. Via the ad-hoc network it finally reaches the server or client component situated at the MN.

From a network topology point of view, this architecture represents a cascaded transfer network. The MN's on-board FLEETNET router acts as an ingress router forwarding IP packets via the ad-hoc network. The FGW provides the corresponding egress interface and acts as an ingress router to the FGW network forwarding packets toward the associated FLEETNET Proxy. Since FGWs usually are connected to the Internet, the FGW network actually constitutes as an overlay network consisting, e.g., of an IP tunnel between each FGW and its associated FLEETNET Proxy. Clearly, other means to establish a FGW network are possible as well, e.g., a dedicated LAN or MAN.

As can be seen from fig. 6.8, the FLEETNET Proxy additionally handles protocol conversions between the VANET and the Internet. The FLEETNET VANET cloud comprises the segment between a vehicle and the FLEETNET Proxy, i.e., it covers the FLEETNET ad-hoc network and the FGW network. Therefore, the FLEETNET Proxy splits up the end-to-end transport layer connection into two communication segments where the leg between MNs and the FLEETNET proxy mainly relies on IPv6 over FLEETNET network layer protocols (cf. sect. 6.3.3).

The FLEETNET Proxy is located in the Internet, e.g., in the domain of an Internet Service Provider (ISP) and should be seen as a virtual instance. It acts as a transparent representative for both the Internet and the VANET: For the vehicles the FLEETNET Proxy represents the CNs, and for the CNs it represents the vehicles. In order to achieve scalability, to avoid bottlenecks and to circumvent a single point of failure, the FLEETNET Proxy may comprise a cluster of FLEETNET Proxies. Moreover, there can be several FLEETNET Proxies acting as transition points between the FLEETNET VANET cloud and the Internet.

In order to scale with the number of vehicles, FLEETNET IP-based communications relies on IPv6. Thereby, every vehicle is identified by a globally unique and permanent IPv6 address from a reserved address space (cf. fig. 6.6). Hence, all vehicles share one common IPv6 'FLEETNET prefix' and, thus, represent a single large IPv6 subnet. Like in typical ad-hoc networking scenarios, there is no hierarchical relation between the vehicles' IPv6 addresses in FleetNet, although each vehicle acts as both an end system and a router. This characteristic results from the following two properties:

- Due to the vehicles' mobility, the topology of the VANET changes permanently. Since a vehicle's global IPv6 address is static, the address structure in the VANET would be permanently reconfigured and an address hierarchy could not be established.
- The location-based forwarding of packets in the FleetNet VANET does not need an IPv6 address hierarchy. Instead is uses the geographical positions of vehicles to deliver the data to the targeted vehicle or to a FGW.

As a result, the FleetNet VANET cloud appears as one common IPv6 subnet sharing the global IPv6 FleetNet prefix and one logical access router, the FleetNet Proxy. If there is more than one FleetNet Proxy, each vehicle is represented in the Internet by exactly one FleetNet Proxy. In this case, the FleetNet VANET cloud also appears as one common IPv6 subnet but with several access routers to the Internet. Thereby, the vehicular ad-hoc network is logically partitioned into several subnets and each FleetNet Proxy forms its own logical subnet. However, in the FleetNet VANET this logical partitioning is not of relevance to enable local IPv6-based communications between the vehicles.

6.3.3 Protocol Architecture

The FleetNet Proxy is responsible for protocol adaptation and conversion between the FleetNet VANET and the Internet. Protocol adaptation is required for both the network layer and the transport layer:

Network layer functions are required to provide interoperability between the Internet (IPv4) and the VANET (IPv6) as well as between IPv4-based user equipment and the IPv6-based on-board network in the context of intra-vehicle communications. Additionally, the network layer must provide IP mobility management for moving vehicles as well as for mobile user equipment in intra-vehicle communications.

The transport layer needs to be optimized for efficiency regarding the characteristics of the VANET, i.e., its fast ad-hoc network reconfiguration and strong variation in availability and reliability of connectivity within the VANET and to the Internet via FGWs. Required protocol modifications and extensions due to this optimization introduce the need for protocol adaptation.

Interoperability issues, mobility management and transport layer adaptation tasks are allocated to the FleetNet Proxy and the FGWs. These components are discussed in detail in the following sections. Intra-vehicular communications requires protocol optimizations within the vehicle only and is outlined further in sect. 6.3.4.

Interoperability

On the network layer, interoperability is an essential condition in order to enable communication between a vehicle and the Internet. In the VANET IP-based communication relies on IPv6, whereas the Internet uses IPv4 technology. The interoperability between IPv4 and IPv6 requires a respective address translation protocol. The FleetNet Proxy uses Network Address Translation – Protocol Translation (NAT-PT) [17]), which provides the most flexibility among the translation approaches [18]. The key concept in NAT-PT is that IPv6-based vehicles specify their destinations using IPv4 addresses, which are embedded into a specific IPv6 address class, the so-called IPv4-compatible IPv6 addresses [19].

The NAT-PT protocol running on the FleetNet Proxy extracts the original IPv4 address and communicates with the targeted Internet host using IPv4. Therefore, NAT-PT requires a certain amount of (global) IPv4 addresses to map the vehicles' IPv6 source addresses onto IPv4 addresses dynamically. An optional feature of NAT-PT is to utilize transport layer identifiers for the address translation, which attenuates the address space restrictions of IPv4. This way, TCP and UDP ports can be used in addition to the IPv4 address, which allows the mapping of several IPv6 addresses onto one IPv4 address.

The interoperability aspect unavoidably re-introduces the address scalability problems coming along with IPv4. Those restrictions must be accepted as long as the Internet does not support IPv6 natively. As the Internet still evolves to IPv6, interoperability should be kept in mind as a necessary migration step toward an IPv6-based Internet. Hence, further discussion will not attend to the interoperability issue any longer in order to improve clarity of the description of basic protocol mechanisms.

Mobility Management

From the discussions in sect. 6.3.1 it becomes clear that Mobile IPv6 cannot be used in the FleetNet VANET without modifications. Thus, FleetNet Internet Integration deploys a mobility management protocol herein denoted as FleetNet Mobile IPv6 (FNMIP6). FNMIP6 was developed with respect to the FleetNet scenario and solves the problem to locate the FGW currently associated with a given MN. It establishes communications between the MN and a CN in the Internet via the FleetNet Proxy using the associated FGW and maintains communications when the MN re-associates with another FGW.

FNMIP6 is based on the principles of Mobile IPv4, but was designed to support IPv6-based mobile nodes organized as ad-hoc networks. A key feature of FNMIP6 is in the fact that communication relies on the global IPv6 addresses of MNs only. FNMIP6 thus consequently avoids link-local and site-local

IPv6 addresses and circumvents IPv6 stateful or stateless automatic address configuration, which conserves bandwidth resources in the ad-hoc network.

In order to manage vehicle mobility, FNMIP6 uses an agent-based system similar to Mobile IPv4 (cf. fig. 6.9). That is, each MN is represented in the Internet by its HA. In contrast to Mobile IPv6, FNMIP6 re-introduces the FA functionality, which represents vehicles in the ad-hoc network with a CoA. This agent-based system is combined with the proxy-based architecture of FleetNet Internet Integration. From a topological point of view, all IPv6 addresses of vehicles are located at the FleetNet Proxy, which also maintains the HAs of the MNs. This way, the FleetNet Proxy acts as a representative of vehicles in the Internet and thus is able to manage the mobility of vehicles. The FA functionality is located at the FGW, i.e., each FGW acts as a FA and represents a vehicle in the FleetNet by associating the vehicle with its CoA.

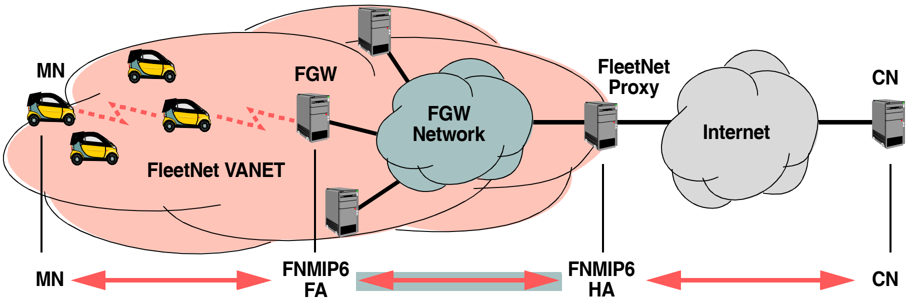


Fig. 6.9. Agent-based system of FNMIP6

The communication flow between a MN and a CN thus follows the scheme listed below (cf. fig. 6.9).

1. If a CN wants to send IPv6 packets to a vehicle, it always directs them to the vehicle's global IPv6 address. This way, the IPv6 packets are routed through the Internet to the FleetNet Proxy that maintains the HA of the vehicle.
2. The HA in the FleetNet Proxy receives the IPv6 packets on behalf of the vehicle and tunnels them to the CoA currently registered for the MN. This way, packets are tunneled to the FGW currently associated with the MN. For the tunneling, FNMIP6 uses either IPv6-in-IPv4 tunneling [20–22] or IPv6-in-IPv6 tunneling [23] depending on the technology supported by the FGW network.
3. The FA on the FGW unpacks the encapsulated IPv6 packets and forwards them to the MN using the geographical multi-hop forwarding of the VANET.

Conversely, a MN that wants to send IPv6 packets to a CN first has to decide whether the packets should be delivered locally to another vehicle within the

VANET or to a CN in the Internet. This can be done easily by comparing the IPv6 prefix of the destination address with the FleetNet prefix, because all hosts in the FleetNet VANET cloud share this common IPv6 prefix. An IPv6 destination address not associated with a VANET node implies the packet to be forwarded to the MN's 'default gateway', that is, the FGW currently associated with the MN.

4. If a vehicle wants to send IPv6 packets to a CN in the Internet, the packets will be delivered to the FGW which maintains the FA the vehicle is currently registered with.
5. In the FGW, the outgoing IPv6 packets are accepted by the FA, which tunnels the IPv6 packets back to the HA using reverse tunneling.
6. Finally, the HA decapsulates the tunneled IPv6 packets and forwards them to the CN addressed in the original IP packet. This way, all IP packets are implicitly routed via the FleetNet Proxy.

Although the use of a FAs in FNMIP6 seems to be a step backwards toward Mobile IPv4, it is an indispensable feature of FNMIP6 for managing IP mobility of vehicles in the FleetNet VANET. A FA entity avoids the need for link-local co-located CoAs for MNs and shifts potential tunneling overhead from the VANET into the Internet. Additionally, FAs enable the deployment of IPv4 tunnels in the FGW network which allows to establish communications between HA and FA via the IPv4-based Internet. The use of reverse tunneling from the FA to the HA is necessary to ensure that all IPv6 packets from a MN to a CN in the Internet pass the FleetNet Proxy. This is mandatory for splitting the transport layer connection, which is described in more detail in sect. 6.3.3.

Discovery and Selection of FleetNet Gateways

If a MN wants to access the Internet it has to associate and register with a FGW, i.e., a FA. Prior to this, a MN needs to discover available FGWs, but, as outlined in sect. 6.3.1, Mobile IPv6 router advertisement/solicitation strategies are not applicable to the FleetNet architecture. In an ad-hoc network, this task usually is handled by service discovery protocols like the Service Location Protocol (SLP), Universal Plug and Play (UPnP), Jini, or Salutation [24].

The traditional service discovery paradigm is reactive, i.e., it is up to the MN to search for available services in the ad-hoc network. A MN therefore sends a service request message to a specific multicast address, which is assigned to all service providers in the network. If a service provider receives such a service request, it will respond with a registration reply message to the MN containing information about the service it provides. As an alternative, the MN sends the service request directly to a predefined service directory ('Registrar'), which manages the available services in the ad-hoc network. However, this reactive service discovery process is harmful in the FleetNet VANET. Each vehicle permanently has to seek its environment for newly available FGWs, so the overhead depends on the number of FleetNet-enabled vehicles. Traditional service discovery is also based on multicast, which is not supported by the FleetNet Network Layer (FNL). Hence, FNMIP6 uses a passive discovery approach to provide a scalable FGW discovery to identify FAs. This approach comprises two basic tasks: the discovery of available FGWs, and the selection of the most suitable FGW.

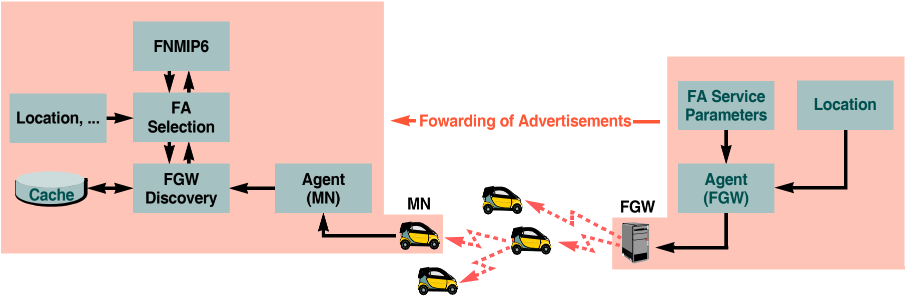


Fig. 6.10. FA Discovery process in FNMIP6

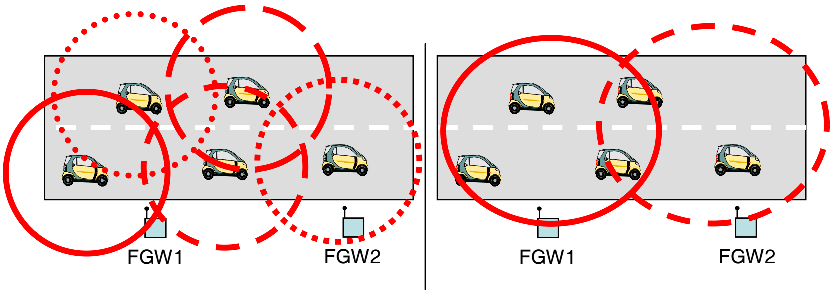
FGWs act as VANET service nodes, that provide FA functionality. For the passive discovery strategy used in FNMIP6, service detection is distributed among FGWs and MNs as illustrated in fig. 6.10.

1. The component located at the FGW periodically emits service advertisement messages providing FGW and service parameters to the component located at the MNs.
2. These service advertisements are distributed by the FleetNet location-based routing algorithm at least within a specified service coverage area of the FGW.
3. The component located at the MNs processes service advertisements received and notifies co-located management components according to the type of service specified by the service parameters received.

The MN caches FGW and service parameters in a database which satisfies subsequent requests from service detection entities locally. Thus, the database reflects the FGWs and their FAs that are currently available. Database entries time out if no service advertisement is received for a given advertisement period. That is, the service has become partially unavailable with respect to local service requests. In fact, database entries are not removed immediately after service advertisement messages fail to appear, since this might be due to a re-organization of the VANET resulting in a temporary loss of connectivity to the selected FGW.

From the vehicles' point of view, the discovery of a FGW is reduced to a search in the local database: If FNMIP6 requires a new FA, it queries the in-vehicle service provider for an available FGW. The service provider searches in the local database and responds with the respective result if the search was successful. Otherwise, FNMIP6 assumes that no FGW is currently available.

Fig. 6.11 visualizes the differences between traditional service discovery and the passive discovery used in FNMIP6. FNMIP6 prevents copious transmissions of service requests from the vehicles. Whereas the number of transmitted messages for service discovery in fig. 6.11 (a) depends on the number of vehicles, the total number of transmitted service advertisement messages in FNMIP6 grows with the number of FGWs. Another very important benefit is, that no multicast traffic occurs in the VANET. Service advertisements are based on geocast only, whereas traditional service discovery approaches require a respective multicast support of the FNL. Furthermore, the overhead of the passive discovery appears in an FGW's coverage area only. In contrast, traditional service discovery protocols permanently have to discover their environment for (newly) available FGWs, which permanently causes overhead in the VANET even if no FGWs are available.



(a) Active Service Discovery (b) Passive Service Discovery

Fig. 6.11. Active vs. passive discovery of gateways

The local database may provide more than one suitable FGW. In such a situation, the 'most suitable' FGW must be selected by the FNMIP6. A number of selection strategies have been investigated differing in complexity, efficiency and suitability for certain traffic and deployment scenarios. This still remains to be a research topic.

The lowest complexity solution applied so far is based on a geographic distance calculation with limited memory (*loiter mode*) and the most complex approach is based on a stochastic system (*fuzzy mode*), which calculates the 'most suitable' FGW in a two-step procedure.

1. The system uses all available information received from the FGW via its service advertisements, such as the current utilization, the number of registered users with the FA, and the position of the FGW. Moreover, local information, such as MN speed, direction or current traffic density is considered.

2. Based on this input, the system estimates the number of users, disconnection probability, bandwidth utilization, and expected packet loss in a first step. These estimates then are used to calculate the suitability of a FGW for specific application requirements in a second step.

Hence, the FGW with the highest rating, i.e., the lowest cost factor, is identified as the most suitable FGW. Details about the selection process can be found in [25].

Mobile Node Association Procedures

If a vehicle newly discovers a FGW it needs to register with the FA and its HA prior to establish communications with a CN. The registration process is based on a request/reply message pair and comprises a four step process (cf. fig. 6.12):

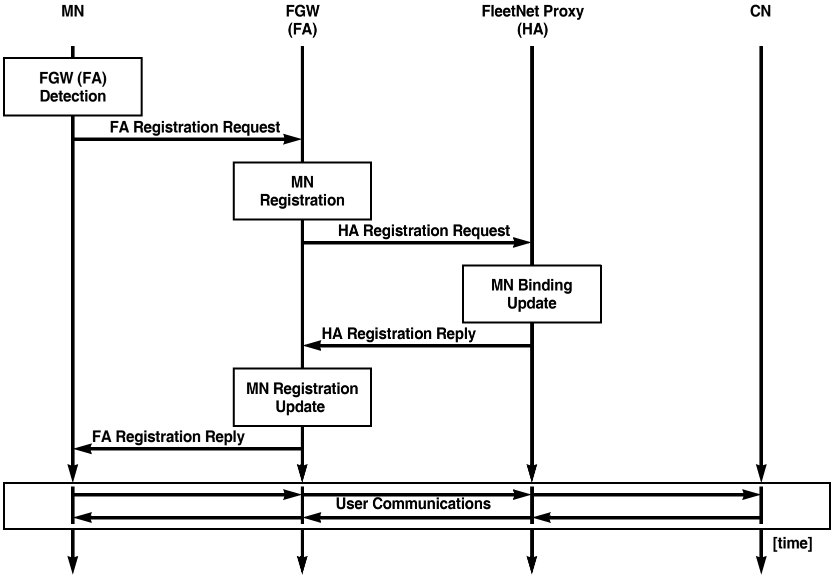


Fig. 6.12. FGW Association (simplified) in FNMIP6

1. The vehicle sends a registration request to the prospective FA to initiate the registration procedure.
2. The FA processes the registration request and updates its internal visitor list. Then, it initiates the MN registration at the HA.
3. The HA processes the registration request by updating its bindings, and responds with a registration reply message to the FA to grant or to deny the request.

4. Finally, the FA processes the HA's registration reply message and relays it to the MN to inform it of the disposition of its request.

After this registration process has been completed successfully, the association with a new FA is established in order to allow subsequent user communications with the CN. This way, the FleetNet Proxy always knows of the current CoA of a vehicle and its associated FGW.

The procedures described so far associate each vehicle with a dedicated CoA. This has been done mainly to avoid modifications not mandatory to the Mobile IPv4/IPv6 paradigms. Since the FleetNet VANET supports native routing of IP packets – due to the well known association of a MN's FNID and its IP address, which implements by a protocol adaptation layer, and the fact that the VANET represents one large IPv6 subnet – all MNs associated with a given FGW can share one CoA which simply is identical to the IP address of that FGW. That is, an IP packet send to a MN is routed via the FGW network toward the FGW currently associated with the MN. If that FGW receives a packet, it knows that the target MN can be reached via its egress interface. It does not need to know, which MN is addressed. Several simplifications may result from this approach:

- The FA co-located to the FGW needs to maintain the association of a MN's global IPv6 address registered with this FGW and the IP address of the FleetNet Proxy responsible for this MN only.
- The HA co-located to the FleetNet Proxy needs to maintain the corresponding information, i.e., the association of MN's global IPv6 address and the Internet Protocol (IP) address of the FGW responsible for this MN.
- For certain FGW network architectures, the FA will degenerate to a managed tunnel endpoint.

Due to the, presumably, smaller number of FGWs in the FleetNet, this will lead to a noticeable reduction of storage size required to maintain registration information especially for low-complexity FGWs, and will speed-up re-association procedures in case of a FGW hand-off. Additionally, no information about the destination MN of a packet can be obtained from the packet header. Encryption of FGW network traffic thus also achieves privacy of communications.

Transport Layer in FleetNet Internet Integration

In the Internet, the Transmission Control Protocol (TCP) provides for a reliable connection-oriented transport service between communicating hosts. The TCP has its origins in fixed networks with low to moderate variations in bit error rates, delay and bandwidth. It was not optimized to operate on top of unreliable mobile networks, since packet loss on the wireless link will be mistaken for a network congestion. Although modern implementations of the TCP respond to such a situation more robust, IP-based communications in the FLEETNET VANET and the FLEETNET Internet Integration bears several unique threats to TCP semantics. For this reason, FLEETNET Internet Integration allows to optionally deploy a dedicated transport protocol implementation denoted herein as the FLEETNET Transport Protocol (FNTP).

As described in sect. 6.3.2, the FLEETNET Proxy splits the end-to-end TCP connection into two segments. Hence, the FLEETNET Proxy communicates with hosts in the Internet using standard TCP, whereas the FNTP can be deployed for communication between MNs and the FLEETNET Proxy (cf. fig. 6.13). FNTP is a counterpart to TCP only, i.e., the FLEETNET Proxy translates between TCP and FNTP. In case of the User Datagram Protocol (UDP), no further modifications are necessary as the unreliable and connectionless semantics of UDP are not violated by the FLEETNET VANET communication characteristics.

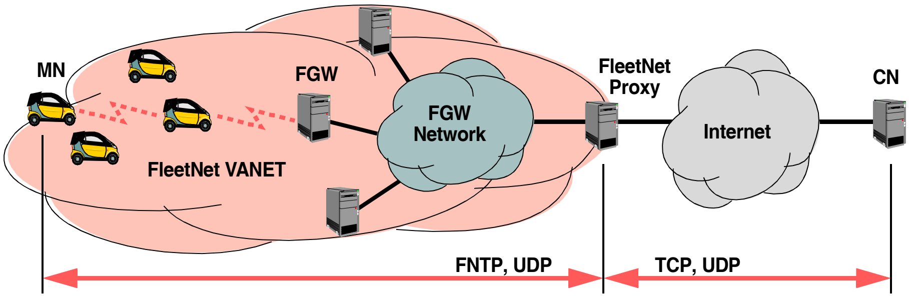


Fig. 6.13. Separation of the transport layer connection in FLEETNET Internet Integration

Current implementations of the FNTP are based on the Ad-hoc TCP (ATCP) approach developed at Portland State University [26]. From a conceptual point of view, the ATCP appears as a sublayer between TCP and IP; fig. 6.14 illustrates the integration in the FLEETNET communication architecture. This sublayer incorporates the complete ATCP functionality by examining ICMP and Explicit Congestion Notification (ECN) [27] messages from the underlying IP layer. The ATCP controls TCP's transmission behavior in case of packet loss, connection breaks, and network congestions. These conditions are very common in the FLEETNET Internet Integration, especially the connection breaks, e.g., when a FGWs becomes temporarily unavailable.

During normal TCP operation mode, ATCP only observes the TCP traffic flow. In case of lossy channels in the VANET, TCP segments might be dropped, forged, or they may arrive out of order. As a result, the receiver generates duplicate acknowledgments for such a TCP segment. If ATCP receives three consecutive duplicate acknowledgments for one segment with an ECN flag, which represents a packet loss due to congestion in the network, it will not deliver the third acknowledgment to TCP. Instead, ATCP forces

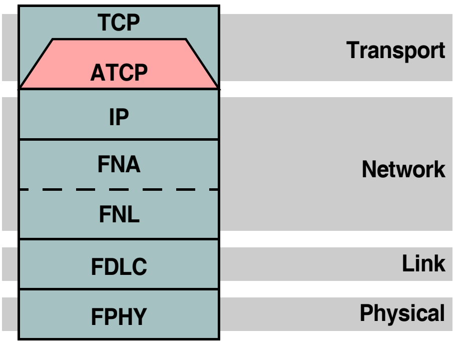


Fig. 6.14. FNTP in the FleetNet protocol stack

TCP to enter the persistent mode. In this mode, TCP is not allowed to transmit new segments. After the ATCP sublayer receives an acknowledgment from the communication peer, it configures TCP to return to its normal operation mode. A similar handling is performed when the VANET gets partitioned and a vehicle becomes disconnected from the Internet. Then, intermediate vehicles along the communication path will generate a 'destination unreachable' ICMP message back to the sender. In this case, ATCP also forces TCP to enter the persistent mode, in which TCP will generate probe packets periodically to test the availability of the connection. If the receiver reconnects, it responds to these probe packets with a duplicate acknowledgment. TCP then moves back into normal operation mode, the congestion window will be reset, and communication will be continued as usual.

In order to 'shorten' the ATCP connection between vehicle and FleetNet Proxy, FNTP may additionally deploy the snoop protocol [28] as illustrated in fig. 6.15. Thereby, a snoop agent is placed in the FGW between the vehicle and the FleetNet Proxy. The snoop agent monitors the packets exchanged through the wireless and the wired connection, which is marked by the right arrow to the snoop agent. It caches each segment that passes the FGW and keeps track of the last sequence number acknowledged by the vehicle. In case of a packet loss on the wireless link to a vehicle, the snoop agent quickly retransmits the respective segment (arrow from the snoop agent to the vehicle in fig. 6.15). This way, the delay for retransmissions is shortened, which improves communication efficiency further on [29].

Protocol Summary

Fig. 6.16 summarizes the mechanisms and communication protocols used in the network layer and in the transport layer of the FleetNet Internet Integration. The IPv6-based network layer consists of the FNMIP6 that implements mobility management. FNMIP6 is deployed in all participating components, i.e. FleetNet Proxy, FGW, and in the MN. The *Mobile Node* implements

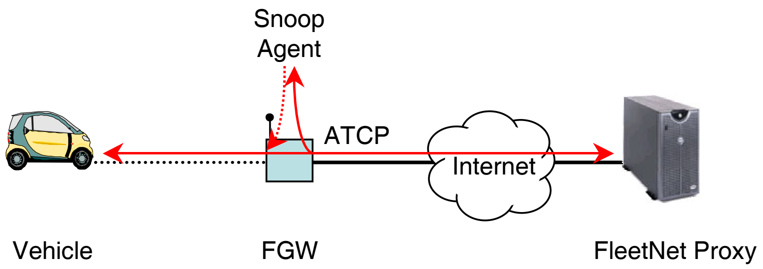


Fig. 6.15. Integration of snoop into FNTP

the mobile part of the FGW discovery (service discovery). The FLEETNET *Gateway* implements the fixed part of the FGW discovery (service advertisement) and the FNMIP6 FA. The FLEETNET *Proxy* implements the FNMIP6 HA. Additionally, it implements NAT-PT to ensure IPv4-IPv6 interworking.

The FLEETNET Internet Integration transport layer consists of the FNTP, which is implemented in the MN and in the FLEETNET Proxy as an optional component. The FNTP provides a more efficient reliable connection-oriented communication in the FLEETNET with respect to standard TCP. Thereby, the FLEETNET Proxy has to translate between FNTP and TCP to ensure the interoperability with Internet hosts.

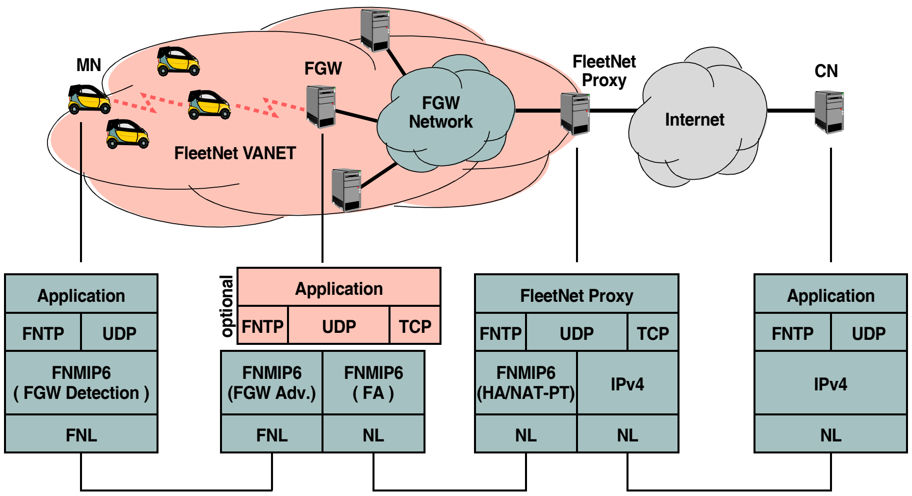


Fig. 6.16. Communication protocol overview in FLEETNET Internet Integration

Some application scenarios of FLEETNET Internet Integration rely on IP-based services provided by FGWs. In these scenarios, a more sophisticated FGW will implement the FLEETNET protocol stack up to the application layer. Besides the MNs, the FGW is the sole component in the FLEETNET VANET architecture, that can implement both FLEETNET-native ('FLEETNET-aware') as well as IP-based ('FLEETNET-non-aware') applications and services. That is, a FGW can translate, e.g., application data received from an Internet-based application to FLEETNET geographical addressing as presented in the chapter about 'The Pinboard Application - Location-Based Messaging for Vehicular Communications', thus providing inherently location-based services.

6.3.4 Intra-Vehicle Communication

Mobility Management and communication protocols described so far enable an efficient data transport between mobile vehicles and fixed Internet hosts. However, it is not possible to support traditional TCP/IPv4-based applications and services located within the vehicles. Besides FLEETNET-non-aware applications running on the communication platform in the vehicle, passengers likely want to use their mobile equipment to access the Internet using the FLEETNET VANET, such as in common mobile office or personal communications applications. These traditional IP-based applications use standard TCP/IP for communications, which cannot be used together with the optimized IPv6-based communication protocols in the FLEETNET VANET.

Passengers' mobile devices might be attached to the vehicle's communication platform via any kind of intra-vehicle network supporting IP, such as MOST, Firewire, Bluetooth^{TM} or IEEE 802.11 (cf. fig. 6.7). In order to ensure interoperability of traditional applications with the FLEETNET VANET, FLEETNET Internet Integration deploys a local proxy within the vehicle, denoted herein as the *Vehicle Proxy*.

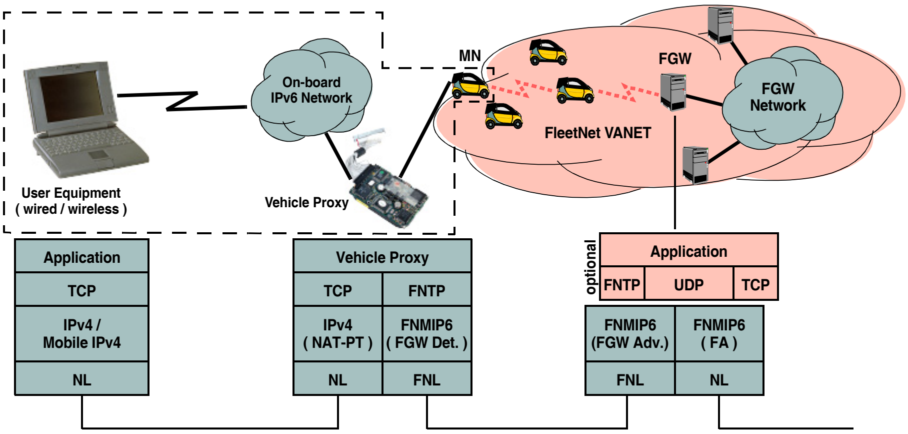


Fig. 6.17. Integration of mobile devices in FLEETNET Internet Integration

The Vehicle Proxy translates back and forth between the FLEETNET communication protocols and the standard Internet protocols as depicted in fig. 6.17. Hence, the overall end-to-end connection between a mobile device within the vehicle and an Internet host is split into the three segments between:

1. Internet host and FleetNet Proxy;
2. FleetNet Proxy and Vehicle Proxy;
3. Vehicle Proxy and the passengers mobile device.

Fig. 6.17 also depicts the communication protocols deployed for supporting legacy applications. The Vehicle Proxy communicates in the FleetNet VANET cloud using the FNTP. Within the vehicle, the TCP is used for communications with the passengers' mobile devices. The Vehicle Proxy thus translates between the FNTP and the TCP the same way as the FleetNet Proxy. On the network layer, NAT-PT performs the transparent translation between IPv4-based intra-vehicle addresses and the IPv6 addresses used in the FleetNet VANET.

The Vehicle Proxy only supports the vehicles' mobility, not the mobility of mobile devices inside the vehicles. This means that FleetNet Internet Integration always delivers the data to the Vehicle Proxy if the vehicle has connectivity with a FGW. In fig. 6.17, a mobile device will usually obtain a private IPv4 address, which is valid within the vehicle only. Hence, the Vehicle Proxy hides the mobile devices inside the vehicle so they cannot be accessed from hosts in the Internet; it is only possible that mobile devices access services in the Internet. In order to ensure an overall mobility, the mobile devices of the passengers have to support Mobile IPv4 as described in sect. 6.2.2.

6.3.5 Communication in a Scattered VANET

As defined earlier (cf. sect. 6.2.1), a mobile ad-hoc network is a transitory association of Mobile Nodes. Especially under sparse MN density conditions, a VANET tends to constantly re-organize into separated ad-hoc networks. That is, MNs situated within a partition are able to communicate, but no communication is possible between MNs situated within different partitions. Such a topology is denoted as a *scatternet*.

In the FleetNet VANET, FGWs implement both a (stationary) MN as well as an Internet-connected host. Thus, it seems feasible to use interconnected FGWs as a means to bridge the gap between separated ad-hoc networks. Since geographic distance has no meaning in the fixed Internet, interconnected FGWs also could be used to span distance in the VANET by reducing the number of hops required in multi-hop communications between MNs.

In the FGW network, communications usually takes place between FGWs and their associated FleetNet proxy only. But, depending on the FGW network architecture, direct FGW-to-FGW communications as well as communications between dedicated instances of the FleetNet Proxy can be achieved easily. This allows for two different options for FGW-to-FGW communications in the FleetNet VANET:

1. FGW-to-FGW via the FleetNet Proxy infrastructure (cf. fig. 6.18);
2. FleetNet VANET multi-hop forwarding via direct FGW-to-FGW communications (cf. fig. 6.20).

Clearly, the first case is a variation of routing between different IPv6 subnets in the FleetNet VANET and applies to IP-based communications only. The second case is an extension to FleetNet VANET multi-hop forwarding and allows to route IP traffic as well as FNL packets.

Scatternet Communications via the FleetNet Proxy Infrastructure

As can be seen from the example in fig. 6.18, MN 1, MN 2 and MN 3 reside in different ad-hoc networks and, thus, are not able to communicate directly via the FleetNet VANET. If MN 1 wants to establish communications with MN 2 or MN 3 it first determines if the destination can be reached within practical limits, e.g., within a given maximum geographical distance. If MN 1 decides to use FGW routing instead of ad-hoc routing, FNMIP6 forwards packets toward the FGW 1 the same way as if an Internet host would have been addressed. FleetNet Proxy 1 decapsulates the packets and forwards them to the targeted destination address, which either is situated within the same IPv6 subnet (MN 2) or outside the range maintained by FleetNet Proxy 1 (MN 3). That is, the FleetNet Proxy treats these IPv6 packets the same way as packets received from the Internet and routes them back into the FleetNet either by FleetNet Proxy 1 toward MN 2 or by FleetNet Proxy 2 toward MN 3.

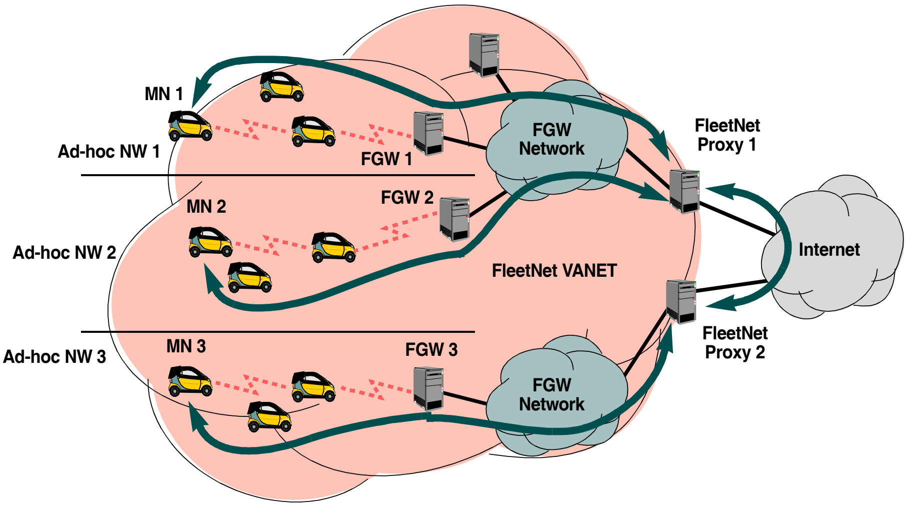


Fig. 6.18. Scattered VANET Communications via FleetNet Proxy Infrastructure

MNs have to decide locally if FGW routing should be used, e.g., based on the process shown in fig. 6.19. The underlying algorithm can be implemented in the FNMIP6 protocol sublayer but needs support from the lower layer protocols, i.e., the FNL management entity, since the location of a destination MN specified by its IPv6 address is not provided to the FNMIP6 by default.

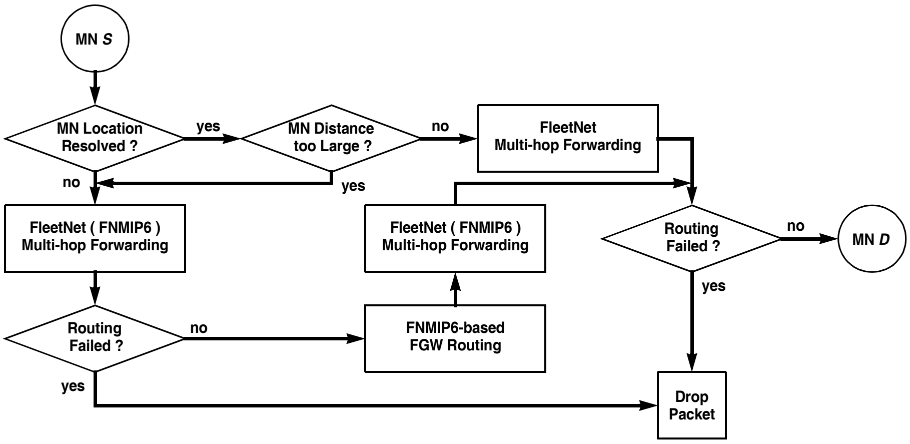


Fig. 6.19. FGW Routing of IPv6 Packets in the FNMIP6-based Approach

Scatternet Communications via direct FGW-to-FGW Routing

The main drawback of the FNMIP6-based FGW routing approach is in its limitation to IP traffic. Additionally, it adds to the traffic load of the FleetNet Proxy if used excessively.

Since all FGWs communicate via the FGW network, the very same infrastructure could be used for direct FGW-to-FGW communications and for routing FNL packets via FGWs without putting a strain on the FleetNet Proxy. This approach has been shown in [30] to integrate seamlessly in the FleetNet VANET.

The example depicted in fig. 6.20 makes clear that direct FGW-to-FGW routing is limited to FGWs that connect to the same FGW network. That is, MN 1 can reach MN 2 via FGW 1 and FGW 2 but not MN 3, since FGW 3 is situated in a different FGW network. Nevertheless, if FGWs are directly connected to the Internet, the FGW network consists of, e.g., a set of tunnels between each FGW and its associated FleetNet Proxy. This topology easily can be extended by a number of tunnels between FGWs.

In a large FGW infrastructure, a fully meshed inter-FGW topology will not benefit too much. Since FGW routing aims at 'bringing a packet closer to its geographical destination by the Internet', a subset of FGWs based on a

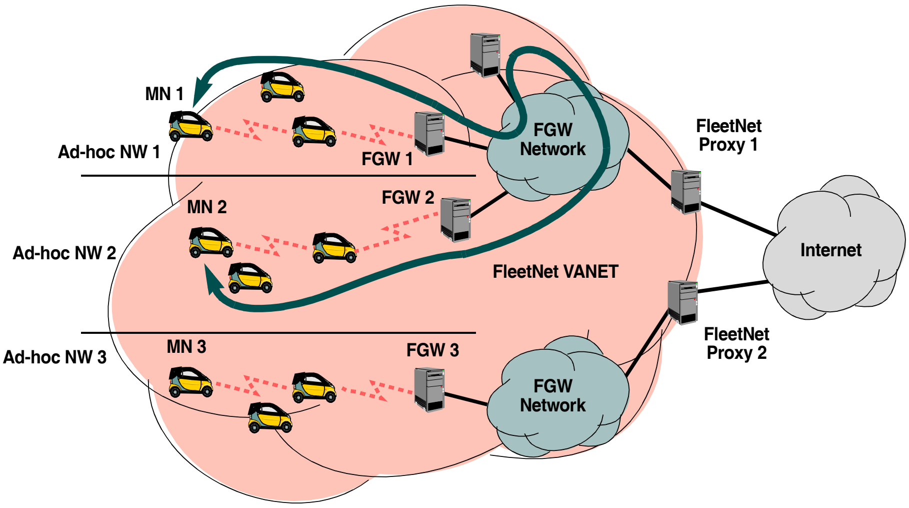


Fig. 6.20. Scattered FleetNet Communications via direct FGW-to-FGW Routing

selection by their geographical location will be sufficient to cover even larger areas. This association can be managed by the FleetNet Proxy.

As shown in [30] this approach scales with the number of FGWs, if a hierarchical organized location service is used for selecting the FGW nearest to the target location of the routed packet. Since each FGW participating in FGW routing needs to maintain a reference to other FGWs including their geographical locations, this will reduce the resource utilization for the per-packet lookup required to determine the target FGW. As an acceptable drawback, this might require routing the packet via intermediate FGWs as can be seen in fig. 6.20.

6.4 Demonstrator

A FleetNet Internet Integration Demonstrator has been set-up as a proof-of-concept and has been tested in a trial in November 2003 in Berlin, Germany.

This trial brought together five vehicles provided by the FleetNet project partners DaimlerChrysler AG and Fraunhofer FOKUS. The gateway infrastructure consisted of four FGWs and two FleetNet Proxies situated at DaimlerChrysler and Fraunhofer FOKUS premises (about 3 km apart). FleetNet Proxy and FGWs have been connected to the corporate Local area network (LAN) of the respective partner forming an FGW network based on IP tunnels between FGWs and associated FleetNet Proxy.

The basic scenarios of FleetNet Internet Integration have been verified in this implementation and found to operate even better than expected. In detail these where

- Access from a vehicle to an Internet host (file transfer and Web access).
- Simultaneous Internet access via the FleetNet and a GPRS.
- Access to the vehicles on-board servers (car data and web server) from an Internet-based client via a FGW.
- Location-based messaging (see the chapter on 'The Pinboard Application - Location-Based Messaging for Vehicular Communications').
- Video streaming from the vehicles' build-in cameras to an Internet-based client.

These tests have been performed in motion, in a multi-hop configuration of up to five vehicles and one FGW at a time. Roaming between FGWs connected to the same FGW network has been verified without loss of connectivity to the Internet. Roaming between FleetNet proxies has been tested but failed to keep Internet connectivity due to the lack of a LAN interconnect.

During development and the integration phase, the vehicles where 'always on' – either in motion or situated in their garage box. This has been very helpful, since software updates, configuration and tests for stable operation of software components could mostly be done via remote access (Secure Shell (SSH), SSH File Transfer Protocol (SFTP)) to the on-board computers. Due to this, TCP support in the FleetNet VANET has been tested extensively and found to work better than expected.

The Demonstrator did not (yet) incorporate mobility support for passengers' personal mobile devices (cf. sect. 6.3.4), FGW routing (cf. sect. 6.3.5) and ATCP support (cf. sect. 6.3.3).

Setting up a demonstrator for the FleetNet Internet Integration was an essential requirement in order to visualize that a VANET can operate as an extension to the Internet. As outlined in the very beginning of this section, FleetNet Internet Integration is of utmost importance to a future market introduction. In this sense, the demonstrator presents a showcase for future developments on applications for VANETs rather than a testing platform and it provides the necessary hands-on experience required to recognize critical issues and to decide on further work items.

6.5 Open Issues, Further Work and Outlook

At the time being, FleetNet Internet Integration has been fully specified. Clearly, there is sufficient room for improvements and it has not yet proved the performance and scalability it has been designed for. Further work thus will focus on a reference implementation targeted as a platform to work on interoperability issues and to carry out performance tests in a real-world scenario.

Some open issues have been identified regarding the service architecture:

- The current FleetNet Internet Integration architecture must be evaluated for options to provide distributed network services to the FleetNet.

Since a FGW is a stationary MN, it might be used, e.g., to provide name or location resolution services to other MNs.

- FGWs might be used to learn about their own infrastructure. That is, in an unmanaged infrastructure, MNs might provide information about gateways received earlier on their way to their current FGW in order to assist other MNs in selecting the optimal FGW.
- A FGW can be seen as a FleetNet VANET service node providing a multitude of services to the VANET or to the Internet. In order to foster development of a gateway infrastructure, the service architecture should be able to share a FGW between multiple service providers. That is, a FGW might be associated with more than one FleetNet Proxy at a time.
- In a commercially oriented environment, the FGW service might be one out of many, e.g., an add-on to a WLAN hot-spot. For this, new interoperability issues, security threats, resource control, or access control requirements may arise and new features might be required.

These issues need to be investigated further since a future market introduction will demand for a clearer picture on development options and business cases. In this, Virtual Private Network (VPN) or closed group communications may come into scope and may demand for solutions regarding access control, resource management, encryption, authentication, or similar to be implemented at the FGWs. This might be in contradiction to the low-complexity approach chosen and may further require the introduction of mobile FGWs.

In order to foster the FleetNet VANET evolution, deployment applications will increasingly come into focus. That is, a bouquet of applications optimized for various stages of VANET deployment and evolution will be required in order to produce a visible benefit from an ad-hoc network in its development stage. It can be assumed, that this type of applications mostly will rely on Internet Integration. This (again) brings up a chicken-egg problem, since network services required might not be available in the phase of evolution that requires this type of application – and vice-versa.

The future role of Internet Integration in vehicular ad-hoc networks is not yet clear. It will be a required feature for the 'connected car', but it seems that it will not come into the view of the driver, passenger or customer the same way as the well known mobile office or personal information applications and services. It rather will be a hidden, ubiquitous or 'silent' feature – providing an indispensable foundation for future vehicular applications but invisibly working in the background.

Acknowledgments

The Authors wish to thank the FleetNet teams at Univ. of Karlsruhe, Univ. of Braunschweig, DaimlerChrysler Telematics Research, and Fraunhofer

FOKUS just to name a few of those that contributed to the vast tasks of bringing FleetNet into the Internet. Special Thanks to Lars Wolf, Thomas Luckenbach and Wilfried Enkelmann for their valuable ideas and their constant support.

Acronyms and Abbreviations

3G 3rd Generation
3GPP 3rd Generation Partnership Project
AAA Authentication, Authorization, Accounting
AP Access Point
ATCP Ad-hoc TCP
CC Call Control
CN Correspondent Node
CoA Care-of Address
DHCP Dynamic Host Configuration Protocol
ECN Explicit Congestion Notification
FA Foreign Agent
FCD Floating Car Data
FGW FleetNet Gateway
FNA FleetNet Network Adaptation Layer
FDLC FleetNet Data Link Control Layer
FNMIP6 FleetNet Mobile IPv6
FNTP FleetNet Transport Protocol
FNID FleetNet Node Id
FNL FleetNet Network Layer
FPHY FleetNet Physical Layer
GGSN Gateway GPRS Support Node (functional entity in 3G mobile networks)
GPRS General Packet Radio Service
GTP GPRS Tunneling Protocol
HA Home Agent
HLR Home Location Register (functional entity in 3G mobile networks)
IAPP Inter Access Point Protocol
ICMP Internet Control Message Protocol
IEEE Institute of Electrical and Electronics Engineers
IP Internet Protocol
IPv4 Internet Protocol Version 4
IPv6 Internet Protocol Version 6
ISP Internet Service Provider
LA Location Area
LAN Local Area Network
MAN Metropolitan Area Network
MM Mobility Management
MN Mobile Node
NAT-PT Network Address Translation – Protocol Translation
PDP Packet Data Profile
QoS Quality of Service
RA Routing Area
RADIUS Remote Authentication Dial-in User Service

RNC Radio Network Controller
SFTP SSH File Transfer Protocol
SIP Session Initiation Protocol
SLP Service Location Protocol
SGSN Serving GPRS Support Node (functional entity in 3G mobile networks)
SSH Secure Shell
TCP Transmission Control Protocol
UDP User Datagram Protocol
UMTS Universal Mobile Telecommunication System
UPnP Universal Plug and Play
VANET Vehicular Ad-hoc Network
VPN Virtual Private Network
WLAN Wireless LAN

References

1. ADASE: Advanced Driver Assistance Systems in Europe. http://www.adase2.net/ (2004)
2. Murphy, A., Roman, G., Varghese, G.: An exercise in formal reasoning about mobile communications. In: Proceedings of the 9th International Workshop on Software Specifications and Design, Ise-Shima, Japan, IEEE Computer Society Technical Council on Software Engineering, IEEE Computer Society (1998) 25–33
3. Banerjee, N., Wu, W., Das, S.K., Dawkins, S., Pathak, J.: Mobility support in wireless internet. IEEE Wireless Communications **10** (2003) –
4. Agere Systems: Inter Access Point Protocol (IAAP) — ORINOCO Technical Bulletin 034/A. Technical report, Agere Systems (2000)
5. IEEE Computer Society LAN MAN Standards Committee: IEEE 802.11F-2003: IEEE trial-use recommended practice for multi-vendor access point interoperability via an inter-access point protocol across distribution systems supporting ieee 802.11 operation. (2003)
6. 3GPP TS 23.234 V2.4.0 (2004-01): 3GPP system to Wireless Local Area Network (WLAN) Interworking; System Description (Release 6). 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects (2004)
7. McCann, S.: 03/458 WIG-Baseline-Document-II. IEEE P802.11 WNG (2004)
8. McCann, S.: 03/459 Interworking-Draft-Skeleton-II. IEEE P802.11 WNG (2004)
9. Rigney, C., Willens, S., Rubens, A., Simpson, W.: Remote Authentication Dial In User Service (RADIUS). RFC 2865, Internet Engineering Task Force (IETF) (2000)
10. Perkins, C.E.: IP Mobility Support for IPv4. RFC 3344, Internet Engineering Task Force (IETF) (2002)
11. Droms, R.: Dynamic Host Configuration Protocol. RFC 2131, Internet Engineering Task Force (IETF) (1997)
12. Johnson, D.B., Perkins, C.E., Arkko, J.: Mobility Support in IPv6. Internet Draft draft-ietf-mobileip-ipv6-24.txt, Internet Engineering Task Force (IETF) (2003)

13. Narten, T., Nordmark, E., Simpson, W.: Neighbor Discovery for IP Version 6 (IPv6). RFC 2461, Internet Engineering Task Force (IETF) (1998)
14. Deering, S., Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, Internet Engineering Task Force (IETF) (1998)
15. Soliman, H., Castelluccia, C., El-Malki, K., Bellier, L.: Hierarchical Mobile IPv6 mobility management (HMIPv6). Internet Draft draft-ietf-mipshop-hmipv6-00.txt, Internet Engineering Task Force (IETF) (2003)
16. Koodli, R.: Fast Handovers for Mobile IPv6. Internet Draft draft-ietf-mipshop-fast-mipv6-00.txt, Internet Engineering Task Force (IETF) (2003)
17. G. Tsirtsis, P.S.: Network Address Translation – Protocol Translation. RFC 2766, Internet Engineering Task Force (IETF) (2000)
18. Waddington, D.G., Chang, F.: Realizing the Transition to IPv6. IEEE Communications Magazine (2002)
19. Hinden, R., Deering, S.: Internet Protocol Version 6 (IPv6) Addressing Architecture. RFC 3513, Internet Engineering Task Force (IETF) (2003)
20. Carpenter, B., Jung, C.: Transmission of IPv6 over IPv4 Domains without Explicit Tunnels. RFC 2529, Internet Engineering Task Force (IETF) (1999)
21. Durand, A., Fasano, P., Guardini, I., Lento, D.: IPv6 Tunnel Broker. RFC 3053, Internet Engineering Task Force (IETF) (2001)
22. Carpenter, B., Moore, K.: Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, Internet Engineering Task Force (IETF) (2001)
23. Conta, A., Deering, S.: Generic Packet Tunneling in IPv6 Specification. RFC 2473, Internet Engineering Task Force (IETF) (1998)
24. Lee, C., Helal, S.: Protocols for Service Discovery in Dynamic and Mobile Networks. International Journal of Computer Research, Special Issue on Wireless Systems and Mobile Computing (2002)
25. Bechler, M., Storz, O., Franz, W., Wolf, L.: Efficient Discovery of Internet Gateways in Future Vehicular Communication Systems. In: Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC), Jeju, Korea (2003)
26. Li, J., Singh, S.: ATCP: TCP for Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications (2001)
27. Ramakrishnan, K., Floyd, S., Black, D.: The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168, Internet Engineering Task Force (IETF) (2001)
28. Vangala, S., Labrador, M.A.: Performance of TCP over Wireless Networks with the Snoop Protocol. In: Proceedings of the 27th IEEE Conference on Local Computer Networks (LCN), Tampa, Florida, USA (2002)
29. Schiller, J.: Mobile Communications. Addison Wesley (2003)
30. Kutzner, K., Tchouto, J.J., Bechler, M., Wolf, L., Bochow, B., Luckenbach, T.: Connecting Vehicle Scatternets by Internet-Connected Gateways. In: Workshop on Multiradio Multimedia Communications (MMC 2003), Communication Technology for Vehicles, University of Dortmund, Germany (2003)

7

Applications for Inter-Vehicle Communication

Wilfried Enkelmann

DaimlerChrysler AG, Research and Technology REI/VI
Alt-Moabit 96A, D-10559 Berlin, Germany,
wilfried.enkelmann@daimlerchrysler.com

Summary. The FleetNet project has aimed at the development and demonstration of a wireless vehicular ad hoc network (VANET) for inter-vehicle communications. Key design factors for VANETs are the capability to distribute locally relevant data where generated and/or needed and to satisfy the vehicle drivers' and passengers' needs for location-dependent information and services. In this chapter we introduce various applications and point out corresponding technical requirements and challenges. Furthermore, we give an overview of the overall system architecture, the experimental hardware and software platform which was used to investigate and demonstrate various applications and mobile information services within a coherent integrated system. The application classes range from cooperative driver assistance to decentralized Floating Car Data services and to user communication and information services. Results obtained from experiments with a fleet of experimental cars will be presented.

7.1 Introduction

Communication into and between cars has attracted major attention during the last few years. Many car manufacturers build multimedia communication devices for the purpose of voice and data communication, for road traffic telematics and for entertainment purposes into their products. For future applications there is a need for a communication *platform.* The FleetNet project has developed a communication platform for inter-vehicle communications based on ad hoc network principles [1].

First efforts to develop an ad hoc network for inter-vehicle communication were spent around 1992 within the project PROMETHEUS [2]. The limited range in the target frequency band at 63-64 GHz and the high costs for equipment at these frequencies have confined this project to a study only and not to a product. Furthermore, at that time the availability of position data could not be assumed to be a wide spread vehicle feature.

Within the European research framework DRIVE, among others, the use of communication technologies was investigated to improve the information distribution into vehicles (traffic telematics). The main focus, however, was the integration of vehicles in cellular mobile radio systems, like UMTS, and broadcast systems, such as DVB-T.

The European Project CarTALK 2000 focuses on new driver assistance systems which are based upon inter-vehicle communication [3]. The main objectives are the development of co-operative driver assistance systems [4] using a self-organizing ad hoc radio network as a communication basis.

In January 2001, the project Inter Vehicle Hazard Warning (IVHW) started [5]. The project was conducted by a Franco-German consortium within the research program DEUFRAKO. The aim of the project is to jointly design and evaluate a common concept for an IVHW system giving precedence to European highway traffic.

More recent work on Roadside-to-Vehicle Communication (RVC) deals with the definition of vehicle-related services by using UMTS as communication infrastructure [6] or by combining UMTS and digital radio broadcast technologies [7]. The distribution of locally restricted information, e.g., emergency messages regarding accidents to inform subsequent vehicles, shall be realized by defining virtual networks of vehicles. The main drawback, however, is that, due to the centralized network topology, the messages have to travel through the air at least twice, from the vehicle to the base station and from the very same base station back to the other vehicles. Additionally, a central unit has to classify and process the messages before further distributing them to virtual vehicle networks. This consumes unnecessary radio resources and introduces delay, which may be decisive for, e.g., vehicles approaching an accident at high speed. In general, the centralized architecture is not efficient with respect to supporting applications, which distribute data only within a group of mobiles that are spatially close to each other. Decentralized architectures are, therefore, better suited for the target system.

The objectives of FleetNet were to develop an inter-vehicle communication platform for different application categories, to implement applications for demonstration purposes, to develop promising introduction strategies [9], and to standardize the solutions found. Multi-hop communications between cars and between cars and stationary network nodes should be provided in an ad hoc radio network by the VANET router (Fig. 7.1).

Before FleetNet was conceived, some components which must be regarded as crucial to such a concept just could not be presupposed. The most important of these are positioning systems. Yet, in the meantime, we can assume that in the mid term, cars will know their positions with an accuracy of about 10 m by using GPS and digital maps, for example. The FleetNet communication platform benefits from this advance in technology by using this information for the organization of the vehicular ad hoc network. First, this information is used for the routing of data traffic on the network level. Second, means of addressing cars based on their positions will be developed, which will enable applications like sending a message to a car running in front or to local floating car data services.

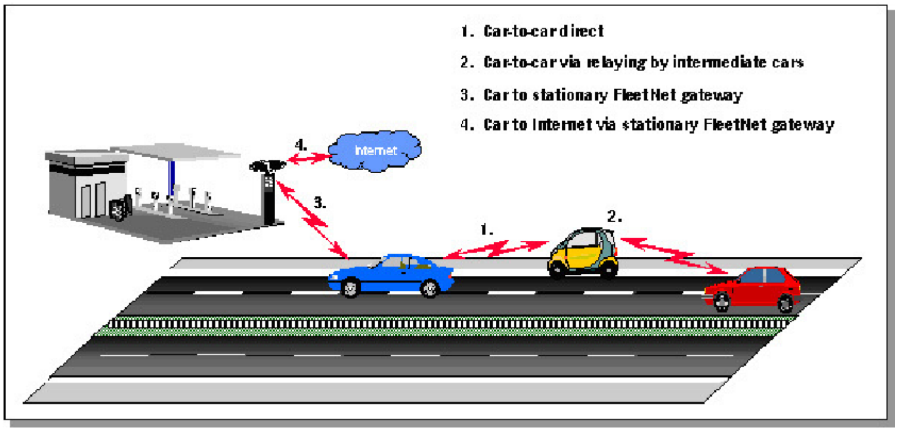

1. Car-to-car direct
2. Car-to-car via relaying by intermediate cars
3. Car to stationary FleetNet gateway
4. Car to Internet via stationary FleetNet gateway
Internet
4.
3.
1.
2.

Fig. 7.1. Communication Scenario.

Based on previous results [8] this chapter describes the experimental system which is used to investigate and demonstrate various applications and mobile information services within a coherent integrated system.

7.2 Applications and Services

VANET services and applications are divided into three categories:

1. cooperative driver-assistance applications (safety-related applications),
2. local floating car data applications, and
3. user communication and information services.

The applications belonging to each class pose their own particular requirements on the communication system in terms of delay, communication range, reliability, positioning accuracy, and bandwidth.

7.2.1 Cooperative Driver-Assistance Applications

Cooperative driver-assistance systems exploit the exchange of sensor data between cars, e.g., exchange of road condition data. One of the first applications which has been implemented was ”emergency notification”. In case of an accident or if the brakes are pressed hard or floored, a notification is sent to following cars. Information of accidents can even be transported by cars driving in the opposite direction and, in this way, be conveyed to vehicles that might run into the accident [10]. Other sub-areas of driver assistance are passing assistance, security distance warning, and coordination of cars entering the same lane. All these applications require position awareness of the vehicles, addressing of vehicles on the basis of their current positions, short transmission delay, and high reliability of data exchange. The bit rate needed to realize these services is low. Cooperative driver-assistance applications provide an excellent example of the need for exchange of data that is of local relevance.

7.2.2 Local Floating Car Data Applications

Current floating car data services are based on service centers which collect and combine data from cars and broadcast the conclusions drawn back to the service members. For example, from several 'no movement' messages of cars on a highway, a traffic jam can be recognized and a traffic jam warning message can be sent to service subscribers [11]. However, such a service can be realized without any centralized information processing in a local inter-vehicle communication system which exploits position-awareness for data distribution, thus avoiding the use of service centers and expensive transmissions via cellular radio systems. Data relevant to cars on the same route can easily be transmitted in the opposite direction of the traffic flow, so that any cars following receive data about the traffic situation ahead. Assuming that cars are equipped with digital maps and are, therefore, aware of the route to travel, messages can be sent along the route to query other vehicles about traffic flow, weather conditions, and other data. Alternative routes can be assessed extremely quickly. The requirements placed on the radio communication system are mean bandwidth, low position accuracy, low data transmission reliability, and medium priority. However, it is expected that data transmission will occur periodically, so that periodical time slots are to be reserved by the channel access scheme [12].

7.2.3 User Communication and Information Services

VANET applications not only deal with the driver's safety and with traffic flow but also with the aspect of comfort. For example, passengers in the back seats can chat or play online games with passengers in other cars traveling on the same highway. Further applications include transmissions of data from commercial vehicles peculiar to their businesses. In contrast to the static presentation of business information prevalent today, i.e., the telephone numbers and logos printed on commercial vehicles, future electronic presentations will be dynamic and can be queried from and transmitted to interested passengers in other cars [13]. With its concept of stationary network gateways on the roadside, VANET also provides a means for marketing along the road. Enterprises can set up stationary network gateways to transmit marketing data to potential customers driving by [14]. Shopping malls or fast food restaurants can inform customers entering their premises about their offers — and can even take orders. With access to the Internet, cars' passengers will receive marketing information while going online via the network gateway.

These services will be based on the Internet protocols. So quality of service requirements are reduced to best-effort service as experienced in the Internet.

The bandwidth requirements of these services are high: the communication range depends on the location of the communication partners. Therefore, it is necessary to control transmission power in order to achieve optimum data throughput in case of different densities of VANET nodes.

7.3 System Overview

One goal of the FleetNet project was to build and maintain an experimental hardware and software platform to investigate and demonstrate various applications and mobile information services within a coherent integrated system. In order to be convincing, people have to look and feel applications and services with an integrated system. Therefore, a system design and implementation was necessary which supports the integration of system components developed by different groups. This enables the identification and investigation of specific aspects with a rapid prototyping approach.

Furthermore, our system includes server and application specific inspection tools which are necessary to perform detailed analysis of the system's behavior. Therefore, we implemented a data logging procedure which stores all relevant information for off-line studies while the applications are running.

Our integrated system consists of the following basic components:

- a group of server modules,
- a group of applications which evaluate information offered by server modules or other applications and
- a router which is the system component that transmits and receives data packets via the communication network.

Figure 7.2 shows the in-car software modules of our system which are installed in each VANET car.

The following server modules encapsulate access to car-specific hardware:

- Module *"navi server"* – a server application that provides the VANET applications with positioning information (longitude, latitude, speed, direction etc.). The application "navi server" obtains its information from a hardware component installed in the vehicle.
- Module *"CAN server"* – a server application that provides the VANET applications with all data accessible from the in-car CAN-bus (speed, milage, light indicator, etc.).

The system is implemented on two PCs. The VANET applications are running under the Windows 2000 operating system whereas the VANET router is implemented on a Linux PC. The VANET applications, router and Showroom are connected via different communication systems. An Ethernet LAN interconnects both PCs. The in-car network is additionally connected to the Showroom via a GPRS communication system. The interfaces of the CAN bus and the navigation system are realized by standard PC Card interfaces.

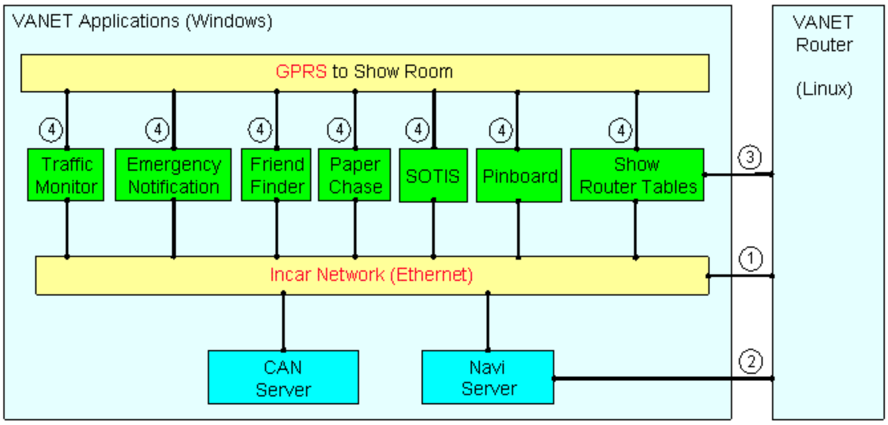


Fig. 7.2. Software modules of the in-car VANET demonstration platform, system overview.

The different connections between software modules of application PC, VANET router, and Showroom are used to transmit the following data (numbers correspond to Fig. 7.2):

1. Link between VANET router and in-car network.
   On this link the VANET data packets are transmitted between both PCs. These data packets are sent to the VANET router in order to transmit them on the inter-vehicle ad hoc network. If the VANET router receives a data packet addressed to himself, he transmits it over this logical link to the in-car network.
2. Link between ”navi server” and VANET router
   Over this link position data provided by the in-car ”navi server” are transmitted to the VANET router. The routing scheme is based on positions. Even more, all VANET nodes maintain a database on the positions of their direct neighbors. To be able to do that, position information of one’s own position have to be forwarded to all neighbors of a vehicle periodically.
3. Link between ”show router tables” and VANET router
   The operation of the VANET routing and forwarding protocols requires maintenance of databases of the neighbors of a vehicle among other control data. In order to display such information in the Showroom, the data have to be transmitted from the VANET router to the in-car network, since the interface between the vehicles and the Showroom is implemented on the application PC.
4. Link between applications and Showroom
   The operation of the Showroom requires position data to be periodically transmitted from the vehicles to the Showroom. In addition to that, gen-

eral application data may be transmitted to the Showroom. The purpose is to display the application results on the global monitor at the Showroom.

This system is installed in a fleet of smart$^{\text{TM}}$ cars to perform experiments. The following Sections describe the implemented applications.

7.3.1 Traffic Monitor

The application ”traffic monitor” (TM)

- collects traffic flow data about the roads on which the cars move and
- displays a digital map with additional information about the current situation.

The traffic monitor requests position data from his navi server which periodically sends current position, speed and other data provided by the navigation system. The traffic monitor generates a TM-message and sends it to the traffic monitor of other VANET members via the VANET router. TM sends data in broadcast mode. A TM-message contains location information as well as static and dynamic attributes of the message generating car.

The traffic monitor map display inside of each car is used to present local information gathered by the distributed VANET. All information accessible inside each car is presented there. This includes it’s own position, the positions of other VANET cars with direct radio connection as well as application specific location-based information.

When the traffic monitor receives a TM-packet from another VANET member the location table and the map display will be updated accordingly. So a complete local view on the current situation will be permanently presented.

Furthermore, the traffic monitor is the basis for a complete global view of VANET which is presented in the Showroom. The main differences between traffic monitor displayed inside the demonstration cars and the Showroom monitor are: all VANET members are connected to the Showroom via GPRS system and the Showroom is not a VANET node. It has no current location info, no network identifier, and is, therefore, not involved in the ad hoc network. The only purpose of the Showroom is to display VANET information in a global view.

7.3.2 Show Router Tables

The task of the software module ”show router tables” is to inspect data concerning the state of the VANET communication system, since the VANET router has no interface to the user. Those data consist of a table with all direct neighbors. The neighbor table will be updated periodically by the VANET router. The module ”show router tables” being executed by the application PC sends a request message to the router which in turn replies and transmits the current neighbor table. Then, the router table information is forwarded to the local traffic monitor and to the Showroom. This module provides an interface which enables a display of data from the VANET router of every VANET member.

7.3.3 Emergency Notification

"Emergency notification" is a VANET application that warns transit motorists of traffic hazards, such as accidents, wet streets or railroad crossings. If the application detects a dangerous situation, a warning message is generated and transmitted to all other nodes that are somehow connected to the sender. Similar to the Inter-Vehicle Hazard Warning System [5] warning messages of the following types will be exchanged over the communication system:

- hazard,
- road works,
- accident,
- stranded vehicle,
- traffic jam,
- attention.

In the current implementation all warning messages will be activated by pressing an emergency button or activating the anti collision light. If a driver detects an emergency event she/he will trigger an emergency message which is transmitted by VANET to the other cars and to the Showroom. The lifetime of the message will be set to a predefined or — depending on the situation — to a dynamically calculated value. When the lifetime is expired the message will be deleted automatically. If another driver detects an already reported emergency situation the lifetime of the message will be reset to the initial value. By additionally sending notifications on the reception of the emergency message to the Showroom, the propagation of the emergency message through the ad hoc network can be shown.

One important feature of the emergency notification will be its ability to provide precise geographically specific alert regions. This ability to precisely warn specific areas by a so called GeoBroadCast-technology is essential to prevent the problems of public information overload. With the proliferation of Global Positioning System technology and the ability to physically transport digital messages by using the oncoming traffic, sending warning messages that contain precise geographic coordinates of the threatened area is feasible. These messages could be transmitted using a GeoBroadCast addressing scheme. Only those VANET nodes inside the threatened region would be informed of the danger. People who do not need to be alerted will not even receive the emergency message. This feature will be included in the next version of the VANET demonstrator.

In order to really warn a driver we have to check which of the received emergency messages is most relevant in the current situation. The emergency message with the shortest distance to our current location is selected as the most relevant message provided that we are approaching the location of the emergency event.

Once the most relevant emergency message is selected we have to decide when to present a warning to the driver. A message will be presented to the driver only if the earliest warning point has been reached. Thus, annoying warnings can be avoided. An approach similar to Nagel et al. [15] is used to calculate the earliest warning point. This approach is based on driver model parameters such as normal and maximum deceleration and driver reaction time. A warning message will be presented accoustically and optically. If normal deceleration is sufficient to stop early enough, additional information such as distance to the event location will be presented ("Accident ahead, in 800 meters."). If we receive an emergency message near to the event location, i.e., we have to break hard, then a more urgent speech output will be given without any additional distance information ("Attention, accident ahead!"). The optical warning message will be displayed until the event location is passed.

7.3.4 Friend Finder

The application "friend finder" shows the possibilities for finding one person in another vehicle with whom an Internet session can be set up. Based on this session different applications can be run. Those applications may include, for example, chat, online games like TicTacToe and paper chase. To be able to set up an Internet application between two cars, at first, each VANET node has to register for the specific application community it wants to join. This registration will be announced to the other VANET nodes. This avoids annoying invitations. If someone wants to start an Internet session a communication peer has to be found. For that purpose a peer from the local list of registered community members has to be selected or a service discovery has to be activated [16]. Then, a data packet which invites the possibly interested passenger in another car has to be sent to the community member (invitation). If the passenger receives an invitation message, he has to reply in a certain time interval in order to accept the invitation. If he is not interested in the requested Internet session he may reject the invitation or just let the acceptance time interval expire. In this context we refer to such a person as a replying person. If the initiator receives an accept message of the replying person he sets up the agreed application. After having enjoyed the Internet session one of the involved persons will terminate the application.

The exchanged data packets in this VANET application use different addressing formats. The first data packet which includes the application registration is addressed to multiple vehicles. Therefore, a broadcast addressing format is used. If an addressee accepts an invitation both vehicles know the network identifier of the other peer. This information is used to address all data packets exchanged in the Internet session.

7.3.5 Paper Chase

In the application ”paper chase” two cars are involved, other vehicles act as forwarders of the data packets. The initiator (follower) starts the application by selecting his partner (leader), with the friend finder application. During this rendezvous dialog the network identifiers of follower and leader will be exchanged. Therefore, data packets will be addressed with unicast. After the selection, follower and leader exchange periodically their position. The task of the follower is to try to follow the leader. As address scheme and communication mode for the position updates, the unicast scheme and unicast mode are used. The objective of the follower is to catch up with the leader. The data messages of the follower are forwarded via multiple hops to the leader. Since the leader sends its position periodically to the follower it is acceptable that a certain amount of messages get lost. In both cars a map showing the last known position of the other car as well as it’s own position will be displayed on the in-car screen. On the Showroom map, both cars and their current position as well as the last known position of the leader from the point of view of the follower will be displayed.

7.4 Experimental results

7.4.1 Experimental Setup

To investigate the practicability of the proposed solution it is necessary to experience applications in realistic environments. Therefore, we have equipped a fleet of `smart`™ vehicles (see Fig. 7.3) with a variety of telematics hardware and the VANET system (see Fig. 7.4). These vehicles are an experimental platform to exploit the characteristics and benefits of the ad hoc network and the applications for inter-vehicle communication. These cars can function both as sender and receiver of VANET messages.

The experimental setup of the demonstrator consists of two main parts:

- a fleet of experimental cars which move on public roads and communicate via the VANET,
- a Showroom which gives a global view of the ad hoc network and the applications.

The technological basis for inter-vehicle communication is at the moment the existing IEEE 802.11b WLAN system with a bandwidth up to 11 Mbit/sec. This allows us to test the integrated system.

We conducted several experiments with our fleet of `smart`™ cars. Experiments with the applications ”emergency notification” and ”paper chase” combined with ”friend finder” are described in the following sections. All results are shown using the map display of the ”traffic monitor”. Furthermore, two more applications developed by partners in the FleetNet project were integrated into the experimental cars:

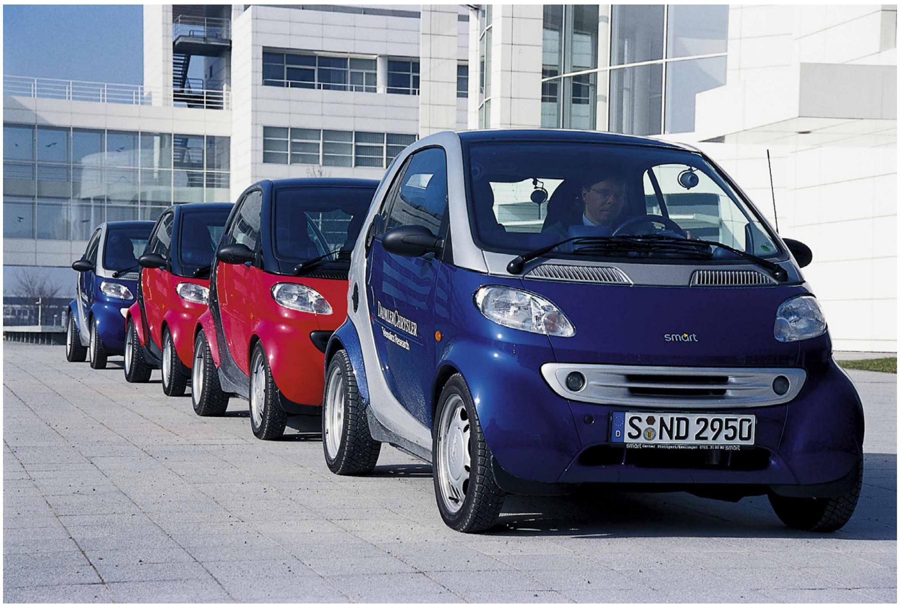


Fig. 7.3. Fleet of smart^{TM} vehicles.

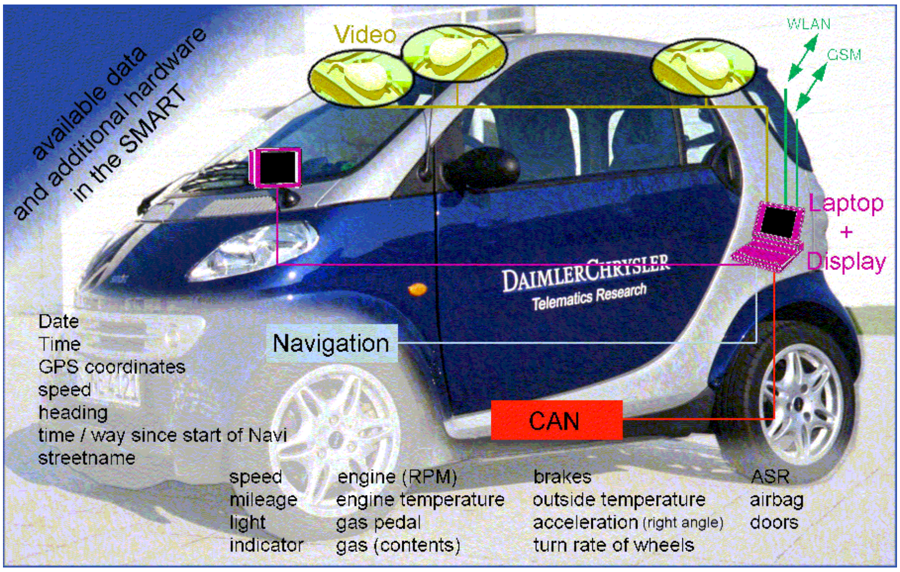


Fig. 7.4. smart^{TM} vehicle equipped with a variety of telematics hardware and the VANET system.

- the application SOTIS (Self-Organizing Traffic Information System [17, 18]) and
- a pinboard application which realizes location-based messaging for vehicular communications [13].

These applications are described in more detail in Chapter 8 and 9 of this book, respectively.

7.4.2 Emergency Notification

We have performed two types of experiments with the application ”emergency notification”, one in a stationary scenario and another in a mobile scenario. In Figure 7.5 the map display of the Showroom is presented. It gives a global view of a stationary scenario. The present position of each car is depicted by a circle. Lines between cars indicate the network connectivity of VANET. The direct neighbors of all VANET cars can be seen in a global view. The location of an emergency message (”traffic jam”) is marked with a traffic sign. The remaining lifetime of the message is indicated by a small clock. The smaller signs indicate that the corresponding cars acknowledged the receipt of the emergency message. Figure 7.6 shows different emergency messages simultaneously.

The in-car display is shown in Figure 7.7. The screen is a control screen which is installed for demonstration and testing purposes. The control view with a digital map shows the local traffic situation. The warning messages which a driver might eventually see are displayed in the user info part of the screen, while the bottom left section is used to generate emergency messages interactively.

In the mobile scenario one car (referred to as smart 1) is located at an intersection. Smart 1 represents a car involved in an accident, it broadcasts an emergency message to all approaching vehicles. This message is displayed in the approaching vehicles in two ways. Firstly, the traffic monitor displays a warning sign at the location of smart 1 when the emergency message has been received (Fig. 7.7). Secondly, an additional warning message is displayed in the approaching vehicle (Fig. 7.8). The latter message is only displayed when the car approaches the accident scene. While leaving the accident scene, the warning message disappears.

The application emergency notification tells us ahead of time that there is a potential hazard ahead of us. It is important to note that this warning message would not be presented to the driver if we travelled the other way. This way unnecessary warnings are avoided. The system can be extended to relay messages even further downstream so that the emergency message is obtained quicker. In this case, a car which receives a message sends this message against the flow of traffic and thus realizes a ”telematic horizon”. Thus, information is transmitted in a region which cannot be surveyed by the driver.

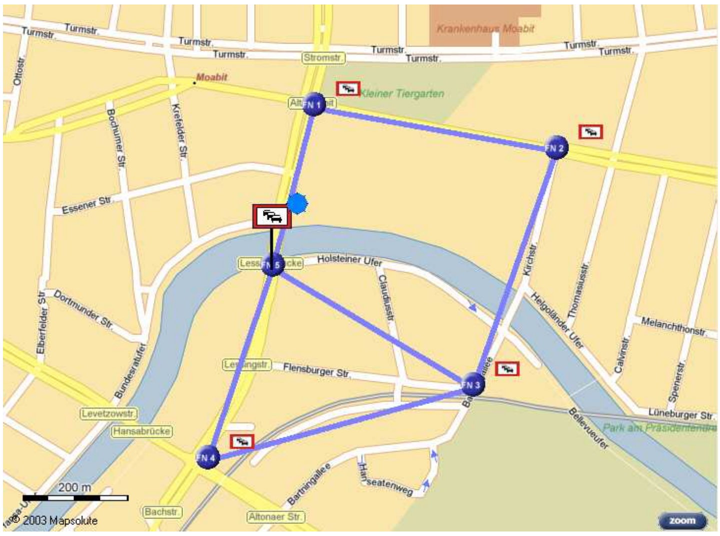


Fig. 7.5. Global view of VANET cars. Network connectivity and an emergency message with corresponding propagation symbols are displayed.

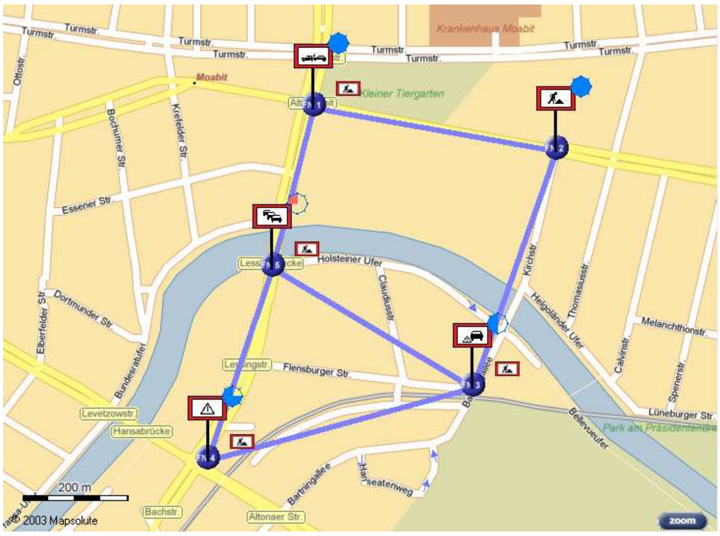


Fig. 7.6. Global view of VANET cars. Network connectivity and different emergency messages are displayed.

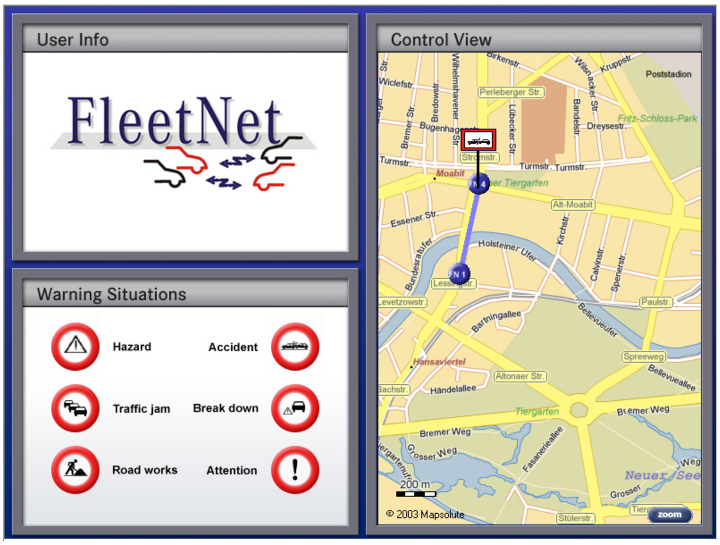


Fig. 7.7. In-car screen with control view, user info, and a button section for generating waning messages. An emergency message "accident" was received which is not yet presented to the driver since the earliest warning point is not reached.

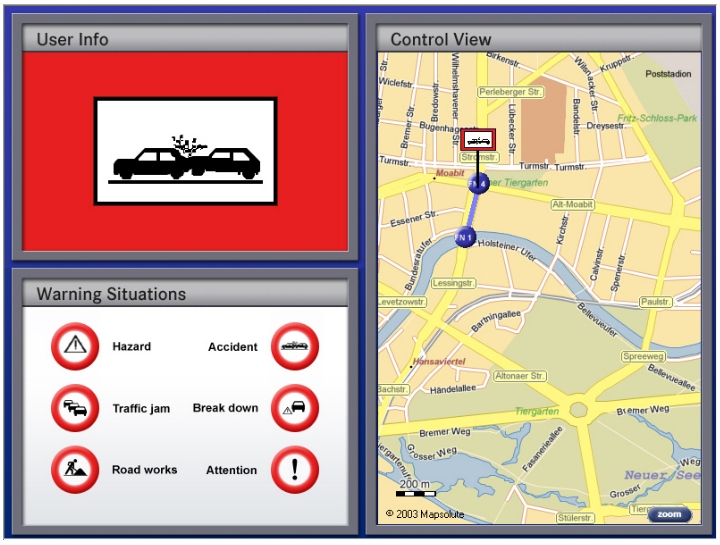


Fig. 7.8. The user info of the in-car screen shows that the emergency message "accident" is presented to the driver since this message is relevant in the current situation.

This application area where communication originates from vehicles can be supplemented by infrastructure-based components which transmit information via stationary beacons.

7.4.3 Friend Finder, Paper Chase

To investigate a scenario where an Internet session between two cars is set up, we conducted an experiment with the applications ”friend finder” and ”paper chase”. We arranged five cars in line so that each car has radio connection to its immediate neighbor (Fig. 7.9). Then, the first and the last car register for the application ”paper chase” indicating that they want to join the community of the specific application. The registered cars are marked with an antenna symbol on the map display.

Then, the passenger of the last car invites a passenger in the first car. This state is indicated with different arrows in Figure 7.10. An outgoing arrow is used for the initiator, an incoming arrow for the invited peer.

If the invited peer accepts the invitation, an Internet session is set up and application specific data are exchanged. In Figure 7.11 this state is indicated with a green and a red circle for the leader and the follower, respectively. The exchanged information, i.e., the current location of the leader, is presented with a flag symbol. This location, the last known position of the leader, is the next destination of the follower. If the follower reaches this location, he will receive a position update from the leader and sets his current destination again to the last known position of the leader. An intermediate state is presented in Figure 7.12. If the follower catches up with the leader (see Fig. 7.13), the Internet session will be terminated.

This application scenario allowed us to study system behavior with simultaneous unicast and broadcast addressing schemes.

7.5 Conclusions

The results obtained with experiments in real traffic situations showed that the wireless ad hoc network for inter-vehicle communications worked effectively. The specified radio protocols for mobile position-aware ad hoc networks were implemented on top of existing radio hardware. Multi-hop communications between cars were provided by means of ad hoc networking. The prototype implementation of applications such as ”traffic monitor”, ”emergency notification”, and ”friend finder/paper chase” demonstrated the feasibility and benefits of such a system.

We have shown that communication allows us to look further away in space and further ahead in time to increase the driver’s information horizon through a new class of applications.

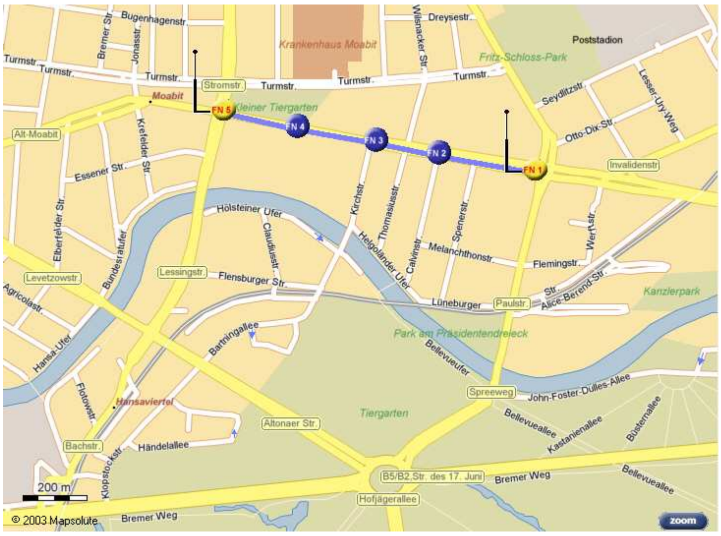


Fig. 7.9. Global view on a convoy scenario. The first and the last car registered for the application "paper chase".

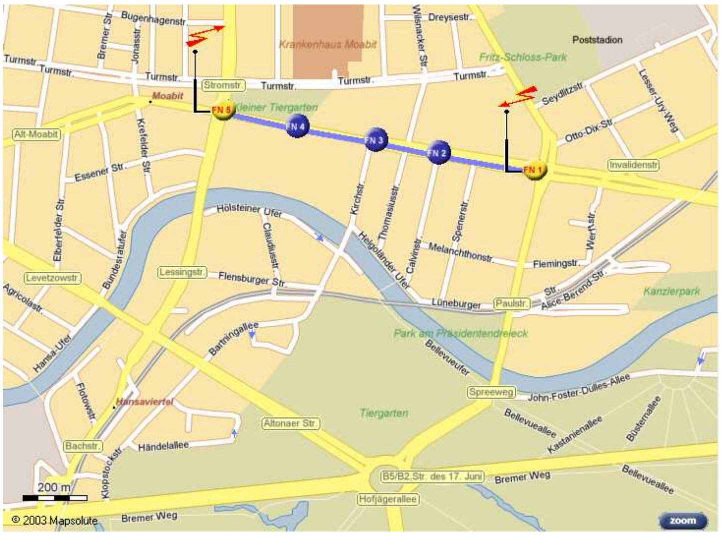


Fig. 7.10. The leftmost car invites the rightmost car for a paper chase session.

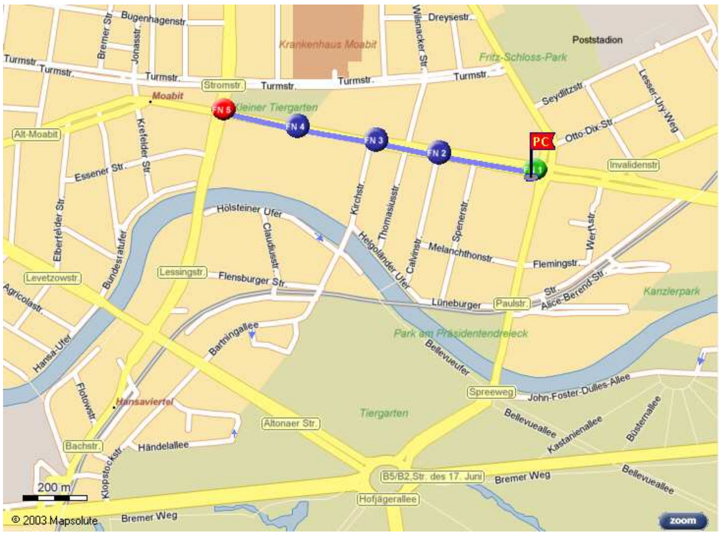


Fig. 7.11. The invited arc accepted to be the leader and sends its current position to the follower. The last known position is marked with a flag symbol.

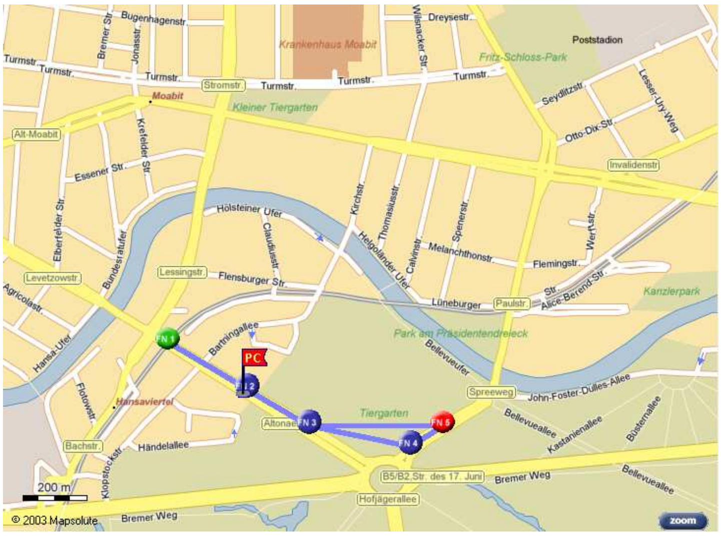


Fig. 7.12. Intermediate state of a paper chase session.

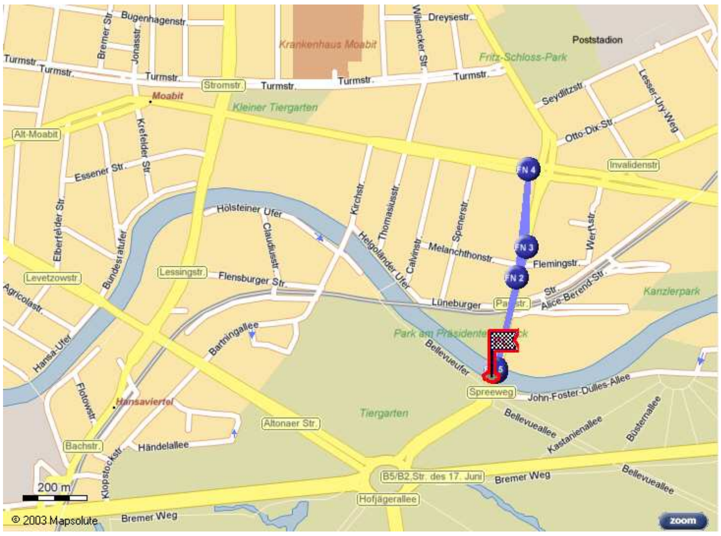


Fig. 7.13. The follower has reached the current position of the leader. This situation terminates the paper chase session.

7.6 Acknowledgements

We would like to thank our FleetNet partners: Fraunhofer FOKUS, NEC Europe Ltd., Robert Bosch GmbH, Siemens AG, TEMIC GmbH, and the universities of Braunschweig, Hamburg-Harburg, Hannover, and Mannheim. I would like to thank G. Matylis for comments on a draft version of this contribution. Special thanks to my students N. Klimin, K. Koulechov, T. Küster, C. Springer, R. Welches, and W. Wieprecht, who contributed significantly preparing and performing the tests with the fleet of experimental cars.

References

1. Hartenstein, H., Bochow, B., Ebner, A., Lott, M., Radimirsch, M., Vollmer, D.: Position-aware ad hoc wireless networks for inter-vehicle communications: The FleetNet project. In: ACM Symposium on Mobile Ad Hoc Networking & Computing, MobiHOC 2001, Long Beach, CA, USA, Oct. 4-5, 2001. (2001)
2. Walker, J.: DRIVE, PROMETHEUS and GSM. In: Mobile Radio Technology, Marketing and Management (COMEX 92), Conference Proceedings, London, UK, 1992. (1992)
3. Reichardt, D., Miglietta, M., Moretti, L., Morsink, P., Schulz, W.: CarTalk 2000 - Safe and comfortable driving based upon inter-vehicle-communication. In: Intelligent Vehicle Symposium 2002, June 17-21, 2002, Versailles, France. (2002)

4. Morsink, P.L.J., Cseh, C., Gietelink, O.J., Miglietta, M.: Preliminary design of an application for communication-based longitudinal control in the CarTalk2000 project. In: e-Safety, Lyon, France, 16-18 Sept. 2002. (2002) Paper No. 2152
5. Chevreuil, M.: IVHW: An inter-vehicle hazard warning system concept within the DEUFRAKO program. In: e-Safety, Lyon, France, 16-18 Sept. 2002. (2002) Paper No. 2016
6. Andrisano, O., Verdone, R., Nakagawa, M.: Intelligent transportation systems: The role of third-generation mobile radio networks. IEEE Communications Magazine **38** (2000)
7. Ericsson: Communication and mobility by cellular advanced radio. ComCar project web site, www.comcar.de. (2001)
8. Enkelmann, W.: FleetNet — Applications for inter-vehicle communications. In: IEEE Intelligent Vehicles Symposium, IV2003, June 9-11, 2003, Columbus, OH, USA. (2003) 162–167
9. Enkelmann, W., Thienhaus, W.: Nutzen dezentraler Ad-Hoc-Kommunikation für vernetzte Fahrzeuge. In: Telematik im Kraftfahrzeug, VDI-Berichte 1728, VDI-Verlag Düsseldorf 2002. (2002) 73–92
10. Briesemeister, L., Hommel, G.: Role-based multicast in highly mobile but sparsely connected ad hoc networks. In: The First Annual Workshop on Mobile Ad Hoc Networking & Computing, MobiHOC 2000, Boston, MA, USA, August 11, 2000. (2000)
11. Franz, W.: Reducing the number of transmissions in floating car data services. In: 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, Sept. 30 - Oct. 4, 2001. (2001)
12. Lott, M., Halfmann, R., Schulz, E., Radimirsch, M.: Medium access and radio resource management for ad hoc networks based on UTRA TDD. In: ACM Symposium on Mobile Ad Hoc Networking & Computing, MobiHOC 2001, Long Beach, CA, USA, Oct. 4-5, 2001. (2001)
13. Welches, R.: Realisierung und experimentelle Untersuchung eines Informationsassistenten im Fahrzeug. Diplomarbeit, Technische Fachhochschule Berlin (2004)
14. Rosen, C.: McDonald's tests wireless payment technology. Information Week, January 8, 2001, http://www.telecomlibrary.com/article/IWK20010108S0004 (2001)
15. Nagel, H.H., Enkelmann, W., Struck, G.: FhG-Co-Driver: From map-guided automatic driving by machine vision to a cooperative driver support. Special Issue on Network, Control, Communication and Computing Technologies for Intelligent Transportation Systems, S.M. Amin, A. Garcia-Ortiz, J.R. Wootton (eds.), Mathematical and Computer Modelling **22:4-7** (1995) 185–212
16. Klimin, N., Enkelmann, W., Karl, H., Wolisz, A.: A hybrid approach for location-based service discovery in vehicular ad hoc networks. In: 1st International Workshop on Intelligent Transportation WIT 2004, Hamburg, Germany, March 23-24, 2004. (2004) 143–147
17. Wischhof, L., Ebner, A., Rohling, H., Lott, M., Halfmann, R.: SOTIS — A self-organizing traffic information system. In: Proceedings of the 57th IEEE Vehicular Technology Conference (VTC '03 Spring), Jeju, Korea, April 22-25, 2003. (2003)
18. Enkelmann, W., Wischhof, L., Ebner, A., Rohling, H.: FleetNet — Anwendungen für mobile Ad-Hoc Netzwerke. Praxis der Informationsverarbeitung und Kommunikation **26** (2003) 197–202

8

Self-Organizing Traffic Information System

Lars Wischhof[1], André Ebner[2], Hermann Rohling[1],
Matthias Lott[3], and Rüdiger Halfmann[3]

[1] Hamburg University of Technology, Dept. of Telecommunications,
Eissendorfer Str. 40, 21073 Hamburg, Germany
EMail: {l.wischhof|rohling}@tuhh.de

[2] Hamburg University of Technology, Dept. of Telecommunications,
now with Audi Electronics Venture GmbH, Vorentwicklung Elektronik,
85045 Ingolstadt, Germany
andre.ebner@audi.de

[3] Siemens AG, Information and Communication Mobile,
Gustav-Heinemann-Ring 115, 81730 München, Germany
{ruediger.halfmann|matthias.lott}@siemens.com

Summary. In an inter-vehicle ad hoc network, the efficient distribution of information is a major challenge. Using Travel and Traffic Information (TTI) as an example, this chapter presents a novel approach for the dissemination of location-dependent information. It is scalable and requires only a very low ratio of all vehicles to be equipped with the communication system – a critical requirement especially during market introduction.

Based on this new technique for data dissemination, a Self-Organizing Traffic Information System (SOTIS) is presented. It provides the functionality of a traffic information system based on a Vehicular Ad Hoc Network (VANET) in a completely decentralized way and outperforms conventional centralized systems in accuracy, availability and cost. The system performance is evaluated using network simulation including vehicular mobility models. It is shown that SOTIS can provide traffic information with a reasonable delay even if only a small fraction of 1-3% of all vehicles is equipped with the system.

Furthermore, a prototype implementation of SOTIS is described. It illustrates the feasibility of the proposed system and allows tests in real traffic situations.

8.1 Introduction

The development of an architecture for inter-vehicle and vehicle-to-roadside communication, such as the FleetNet platform, is motivated by attractive new services – for the driver of a vehicle as well as for the passengers. For a successful market introduction, applications and basic communication methods need to be developed which offer a real benefit for the user, even if only a low rate of all vehicles (e.g. 1-3%) is equipped with the proposed communication system.

With such low penetration rates, successful information dissemination over distances larger than the radio transmission range is a real challenge, since conventional multi-hop communication is very unlikely to be successful in this case.

Therefore, the contribution of this chapter is twofold: first, suitable methods for efficient information dissemination are presented, which require only a minimal penetration of 1-3%. Second, based on these methods, an attractive example application is presented: a Self-Organizing Traffic Information System (SOTIS). Accurate and up-to-date traffic information for the local area is a service highly demanded by vehicle drivers. Current traffic information systems rely on a centralized structure and have severe disadvantages, including limited availability (restricted to highways), high delays and communication costs for the end user (for GSM/UMTS based services).

The chapter is organized as follows: Section 8.2 gives brief a overview of related work. In Section 8.3, a novel approach for information dissemination in vehicular ad hoc networks called Segment-Oriented Data Abstraction and Dissemination (SODAD) is described. Its benefits are illustrated in Section 8.4, where it is applied to the SOTIS application. The performance of SOTIS is evaluated in Section 8.5. The basic system is extended in Section 8.6 with an adaptive broadcast mechanism, which significantly reduces the required data rate. A prototype implementation of SOTIS is presented in Section 8.7. Section 8.8 concludes the chapter with a short summary.

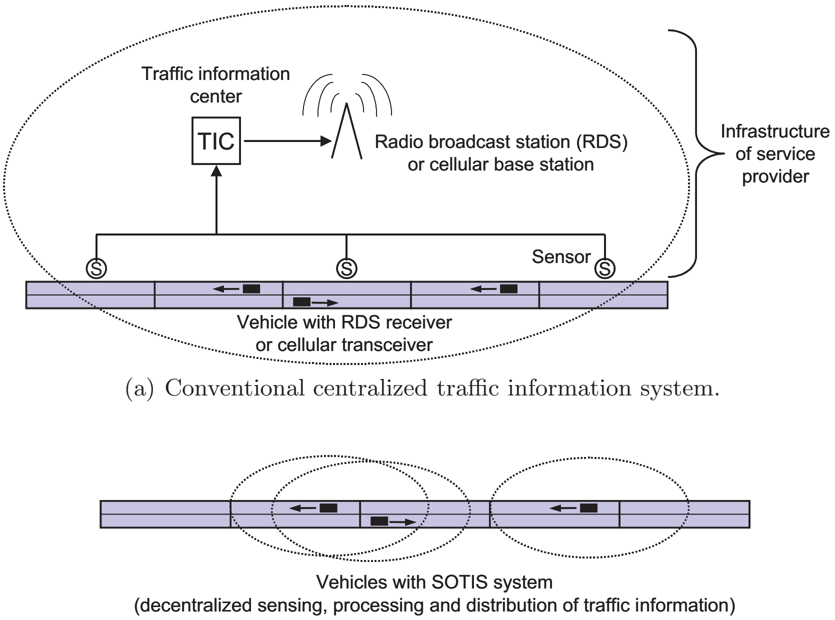
8.1.1 Motivation and Basic Idea

Conventional Traffic Information Systems (TIS) are organized in a centralistic way as illustrated in Figure 1(a): Sensor-based traffic monitoring systems deployed directly at the roadside collect information about the current traffic conditions. This data is transferred to a central Traffic Information Center (TIC), where the current road situation is analyzed. The result of this situation analysis is packed into messages for the Traffic Message Channel[4] (TMC), forwarded to the FM radio broadcast station and transmitted via Radio Data System (RDS) to the driver. Alternatively, the traffic messages can be transferred on demand via cellular mobile phone network.

A centralized service for distributing traffic information has several technical disadvantages:

- A large number of sensors needs to be deployed since the traffic information service is limited to streets where sensors are integrated. Thus, a large investment for the communication infrastructure (sensors, central unit, wired and wireless connections) is necessary. As a result, accurate traffic information is currently not available within cities, since the investment for integrating sensors in city streets would be too high.

[4] The Traffic Message Channel (TMC) is an application of the FM Radio Data System (RDS) for broadcasting traffic information.


Traffic information center
TIC
Radio broadcast station (RDS) or cellular base station
Infrastructure of service provider
Sensor
S
Vehicle with RDS receiver or cellular transceiver
Vehicles with SOTIS system (decentralized sensing, processing and distribution of traffic information)

(a) Conventional centralized traffic information system.

(b) Decentralized self-organizing traffic information system.

Fig. 8.1. Comparison of the conventional form of traffic information systems with the proposed SOTIS.

- The recorded traffic density data is transmitted for traffic analysis to a central unit (TIC). This procedure causes a relatively high delay (typically 20-50 minutes), before the result is broadcasted to the drivers.
- It is not suited for time-critical messages, e.g. emergency notifications.
- Since a central unit covers a relatively large area and due to the limited bandwidth[5] for transmitting the traffic messages, only major events are transmitted. A constantly updated and detailed information for the local area is not available.
- In case of cellular distribution of traffic information, service charges will apply.

For all these reasons, an alternative and completely different approach for monitoring the traffic situation and distributing the traffic messages to vehicle drivers is proposed: the Self-Organizing Traffic Information System (SOTIS) [1], which is based on a Vehicular Ad Hoc Network (VANET). This decentralized self-organizing traffic information system is designed by combining a digital map, a positioning system (e.g. GPS) and wireless ad-hoc communication among the vehicles. Since the first two components are al-

[5] In case of the TMC, the data rate for sending traffic messages is limited to 37 bits/s.

ready available in modern vehicles equipped with navigation systems, the only additional requirement is a simple wireless interface for Inter-Vehicle Communication (IVC). In this decentralized self-organizing traffic information system (SOTIS), vehicles inform each other of the local traffic situation by IVC as illustrated in Figure 1(b). The traffic situation analysis is performed locally in each individual car. There is absolutely no communication/sensor infrastructure needed.

Basically, the communication among the individual vehicles in such a decentralized traffic information system could also be achieved by using a cellular network, such as GSM or UMTS. However, IVC has two key advantages:

– **No service fees:** IVC requires no communication infrastructure or service provider. Service charges are completely avoided – except for the relatively low initial cost for the IVC system, communication is provided free of charge for the end user. This issue is especially important for the considered service of a traffic information system: Although relatively large amounts of data need to be exchanged in order to offer a constantly updated view of the local area, the users' willingness to pay for such kind of service is limited.
– **Direct communication:** Since the vehicles communicate directly without any intermediate base stations, the communication delay is lower compared to centralistic systems. In the vicinity of the cars, the delay is even very low. In contrast to cellular systems, IVC is also suitable for the distribution of extremely time critical data such as emergency notifications in the area of an accident. Furthermore, vehicles can communicate even in rural areas not covered by cellular systems – the communication network is established by the cars themselves and available everywhere.

8.1.2 Technical Challenges

All applications that make use of IVC face two major challenges: required penetration and scalability. The first becomes most obvious for safety applications, where in general it is assumed that a vehicle in an emergency situation transmits a warning message to an approaching vehicle. In the case of low market penetration, e.g. when the system is being introduced, the chances are low that both vehicles are equipped and the second vehicle can be warned. Many comfort applications face a similar problem: With low market penetration, in the majority of the time there is no or only a very limited number of partners in transmission distance. Therefore, the average range in which information can be distributed by (multi-hop) communication is small. This is illustrated in Figure 8.2, where the average multi-hop range for a typical scenario is shown when penetration and transmission range of the system are varied. A penetration rate of 25 % is required to achieve an average multi-hop range of 10 km in a typical highway scenario. Such a high ratio of equipped vehicles will not be available when a IVC system is being introduced.

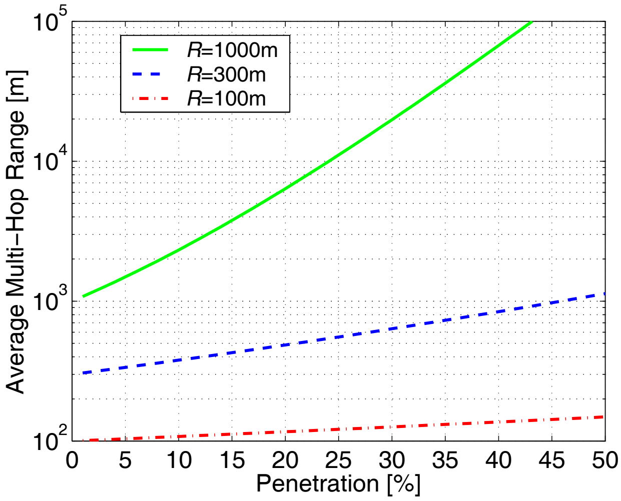


Fig. 8.2. Average multi-hop range for a 4 lane highway scenario with a traffic flow of 450 vehicles/h/lane and an average velocity of 120 km/h. Here, R denotes the assumed transmission range of the air interface.

Scalability becomes an issue once a higher market penetration is reached. In order to avoid overload conditions, the data transferred needs to be restricted. Therefore, some method of analyzing and abstracting the transmitted data is necessary. For applications that require the addressing of individual vehicles, an efficient location lookup service is also required.

In order to solve these two challenges, we propose Segment-Oriented Data Abstraction and Dissemination (SODAD). SODAD can be used to create a scalable decentralized information system that can provide data in an information range multiple orders of magnitude larger than the transmission range of the air interface even if only 1-3 % of all vehicles are equipped with an IVC system. Using the SODAD approach for information dissemination, the novel Self-Organized Traffic Information System (SOTIS) is able to offer very detailed traffic information for the local area of a vehicle while still requiring only a very low market penetration.

8.2 Related Work

The application of digital radio techniques for in-vehicle active security and safety systems is relatively new: In [2], a simple multi-hop broadcast technique for the distribution of traffic information generated by a vehicle is proposed. Packets received from surrounding vehicles are forwarded in order to extend the information range beyond the transmission range of a single vehicle. This flooding mechanism is limited to a few hops by using a hop counter inside each packet. Additionally, in [3], a layered data structure is used, which allows a forwarding node to reduce the packet size by discarding non-relevant information. The idea is to exploit the fact that the needed accuracy of (traffic) information is distance-dependent, which is done similarly in SOTIS.

In [4], a multi-hopping algorithm for IVC is presented, which also uses a hop-limited flooding scheme. The algorithm makes use of position information available via GPS: Before forwarding the message, a receiver determines a waiting time that is anti-proportional to the distance to the sender – therefore, larger hops are favored in the forwarding process, which increases the efficiency. A similar method is used by the TrafficView [5] system developed in the e-Road project. It monitors the location of nearby vehicles in order to extend the view of the driver using IVC. Position information for the individual vehicles is distributed by a broadcast technique.

A peer-to-peer collision warning system [6] based on IVC has been developed by Ford Research. Vehicles broadcast their current parameters such as location and velocity. Using information received from vehicles in the local area, potential hazards are predicted and the driver is warned in advance.

However, all the approaches mentioned above have in common that they are targeted at the distribution of emergency or traffic jam information within an area relatively close to the vehicle by using a flooding approach. The influence of very low ratios of equipped vehicles and the scalability necessary to achieve information ranges of 50-100 km, which are the two central aspects of the methods proposed here, are not considered.

8.3 Segment-Oriented Data Abstraction and Dissemination

Segment-Oriented Data Abstraction and Dissemination (SODAD) is an algorithm for the efficient information distribution in inter-vehicle networks. It is suitable for a wide range of applications in this area, e.g. traffic information systems, parking-lot, motel or gas price information, peer-to-peer travel information and many others. This section describes the SODAD algorithm and its basic ideas.

8.3.1 Map Based Data Abstraction

Data distributed in a vehicular information system typically has the following properties:

- It has a spatial component, e.g. since it describes the situation at a specific location.

– The relevance for a receiver decreases with increasing distance to the location where the data was originally generated.[6] This also means that delay and reliability constraints become more relaxed with increasing distance.

These properties are exploited in SODAD: It is assumed that each vehicle is equipped with a digital map. The map is divided into segments of a known length, which can vary based on the type of object (e.g. road) that is considered.

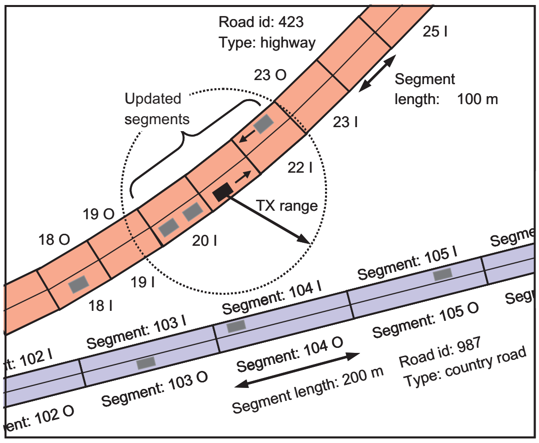


Fig. 8.3. Example for map based data abstraction by segmentation of the roads in the local area.

Figure 8.3 shows an example where a vehicle driving on a highway chooses the road segment length automatically and adaptively: A segment length of 100 m is chosen for the highway where the car is driving on and a larger segment length of 200 m is selected for the country road. The reason is that the highway is considered more important. Due to the digital map and a standardized selection of the segment size, each segment can be identified by an unique identifier, e.g. the combination of road id plus segment number plus direction.

Each car generates new information for all segments in transmission range. This is done either by sensing the information itself or by receiving information observed by other vehicles. In the data abstraction process, a function is applied to all K information elements $I_{s,k}$, $k \in [1 \ldots K]$, received for one segment s. The nature of this function depends on the application, for example

[6] For example, in general the interest of a driver in price information of a gas station nearby is higher than for a gas station 100 km away.

the mean over all $I_{s,k}$ can be calculated or the maximum can be chosen. The result becomes the new data value for segment s and is stored onboard the vehicle.

This process guarantees the scalability of the information system: The amount of data distributed is independent of the number of equipped vehicles and received data messages. The network load depends only on the segment length and the area to be covered. However, a higher number of equipped vehicles improves the accuracy and decreases the delay with which data is distributed – as outlined in the following sections.

8.3.2 Data Dissemination

The second part of SODAD is the dissemination of the per-segment information by using the wireless link. The main objective is that data distribution over large distances is achieved, even in cases of low penetration or low density of vehicles. Therefore the wireless communication is based on two principles:

1. **Local Broadcast:** All data packets are transmitted in form of local (1-hop) broadcasts. Nodes are never directly addressed and no routing of data packets in the traditional sense (i.e. on the network layer) is performed. This exploits the fact that data sensed by a vehicle at a specific location is usually relevant for all or many vehicles in direct transmission range. Furthermore, since no routing is required, there is no overhead for a location service or setting up a route to a specific destination.
2. **Application Layer Store-and-Forward:** Since all data is sent in form of one hop broadcasts, the application is responsible for forwarding the per-segment information. Information received at a node is always analyzed and compared with the currently available information. Only if it is still relevant and more accurate than the previously known information, it will be included in the next broadcast packets. The application recurrently sends broadcast packets with (parts) of its current information on all relevant segments in the local area. More important segments are included more often.

The effect of this communication paradigm is shown in Figure 8.4, which illustrates the communication in situations with a low density of equipped cars: In 8.4 a), Vehicle A and Vehicle B are driving in the same direction. Vehicle B senses the conditions ahead of Vehicle A but since the distance $D(A, B)$ between the two vehicles is much larger than the transmission range D_{TX}, they cannot communicate directly. Later, in 8.4 b), Vehicle C on the opposite lane is in transmission range of Vehicle B. It receives and stores the broadcasted per-segment data. In 8.4 c), the vehicles travel for a while without any communication partner in range. Finally, in 8.4 d), Vehicle A can receive the information from Vehicle B although both vehicles were never in (single- or multi-hop!) communication range.

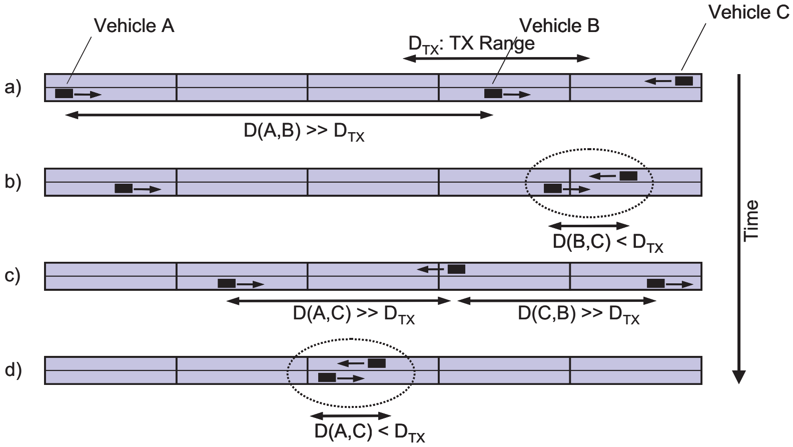


Fig. 8.4. Data distribution in cases of low penetration.

8.4 Self-Organizing Traffic Information System (SOTIS)

As outlined in Section 8.1.1, the basic idea of SOTIS is to provide the functionality of a traffic information system in a completely decentralized way. In the following, the basic algorithms and the technical structure of SOTIS are presented.

8.4.1 Application of SODAD

SOTIS is a typical case where the application of Segment-Oriented Data Abstraction and Dissemination (SODAD) is advantageous. All the requirements outlined in Section 8.3.1 are fulfilled. For each road segment that a vehicle drives, it records the observed average velocity. The data value I_s for a segment s is the mean of the own velocity $I_{s,0}$ and that of all K other vehicles in transmission range:

$$I_s = \frac{1}{K+1} \sum_{k=0}^{K} I_{s,k}, \quad K \in \mathbb{N} \cup \{0\}. \tag{8.1}$$

Additionally, a time stamp indicates when the information for a segment was last updated. Thus, only one data value and one time stamp per road segment are recorded. This sufficiently characterizes the current traffic situation in this segment, since the road type is also known. Within the system, this per-segment information is distributed using the technique known from Section 8.3.2. Recurrently, the vehicles broadcast the available information. Each vehicle obtains traffic information for all road segments in the local area. This information can either be displayed to the driver (e.g. by coloring the roads displayed in the navigation system according to their traffic conditions) or it is used for calculating the best (i.e. fastest) traffic route for the current situation. Furthermore, if critical changes in the traffic situation occur (e.g. an accident), an emergency message is instantly transmitted to all vehicles in transmission range. In this special case, a small data packet including the exact position and type of the emergency is sent.

8.4.2 SOTIS System Structure

The functional structure of the SOTIS components in each vehicle is illustrated in Figure 8.5: Traffic information received either from other vehicles via the wireless link or gathered by the vehicle itself is collected in the knowledge base. It contains the traffic information for all segments in the local area of the vehicle. Information is discarded if it has become outdated or the capacity (determined e.g. by the available memory) of the knowledge base is reached. Using the information stored in the knowledge base, a traffic analysis is continuously calculated in each car. Detailed traffic information is instantly available, for example in order to calculate a dynamic route to a destination. Furthermore, the analysis determines the information to be included in the next broadcasted data packet (examples for selection criteria are presented in Section 8.7). By adapting the rate for generating these recurrent broadcast packets to the local conditions[7], overload situations can be completely avoided.

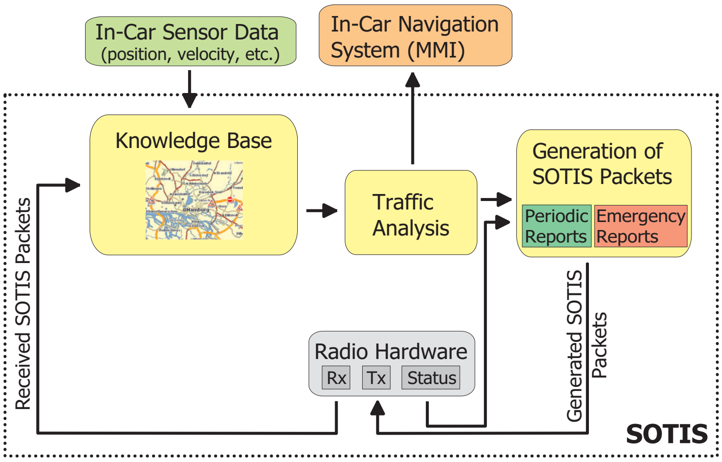


Fig. 8.5. SOTIS system implemented in each vehicle.

[7] This aspect is covered in more detail in Section 8.6. Until then, a non-adaptive system is assumed.

8.5 Performance Evaluation of SOTIS

In order to evaluate the performance of SOTIS, a simplified model has been developed which was implemented within the network simulator[8] *ns-2*. The simulation and system parameters as well as necessary modifications or extensions of the simulator to integrate SOTIS are outlined in the following.

8.5.1 Road Traffic Simulation and Parameters

For simulation, the road traffic and movement of individual nodes (cars) must be described by a mathematical model. In this respect, the typical movement pattern in an inter-vehicle ad-hoc network is very different compared to a general ad-hoc wireless network simulation (where schemes like the "random walk" model can be used to simulate node movement). Therefore, the ns-2 simulator has been extended with a movement model based on a microscopic traffic simulation using a cellular automaton approach [7]. It also allows passing, if safety conditions are not violated [8].

For a each simulation, a road pattern is created (usually based on a map of a highway or city section). The scenario presented in this chapter simulates a regular highway situation with 2 lanes per direction. Table 8.1 lists the parameters used for road traffic simulation. Arrival times are assumed to be Poisson distributed, initial time gaps between adjacent vehicles are therefore chosen from an exponential distribution.

Table 8.1. Parameters used in the traffic simulation.

Road length	140 km and 110 km
Number of lanes	2 per direction
Deceleration prob.	0.4
Constitution of traffic	15 % slow, 85 % regular vehicles
Desired velocity	108 km/h (slow), 142 km/h (regular)
Avg. headway	2 s, 3 s, 4 s (exponentially distributed)
Number of vehicles	≈ 7500, ≈ 10000, ≈ 15000
Mean Velocity	95.6 km/h, 101.3 km/h, 106.4 km/h

8.5.2 Communication Parameters

The link layer assumed for the simulations is a standard IEEE 802.11 system with a data rate of 1 Mbit/s. All packets are transmitted as broadcast. The transmission frequency is set to 2.472 GHz, transmission power is 15 dBm. The receive threshold is adapted in order to achieve a communication range

[8] `http://www.isi.edu/nsnam/ns/`

of 1000 m, since this is the expected range of air interfaces developed for IVC, e.g. DSRC or UTRA TDD Ad Hoc [9]. Omni-directional antennas at a height of 1.5 m are assumed.

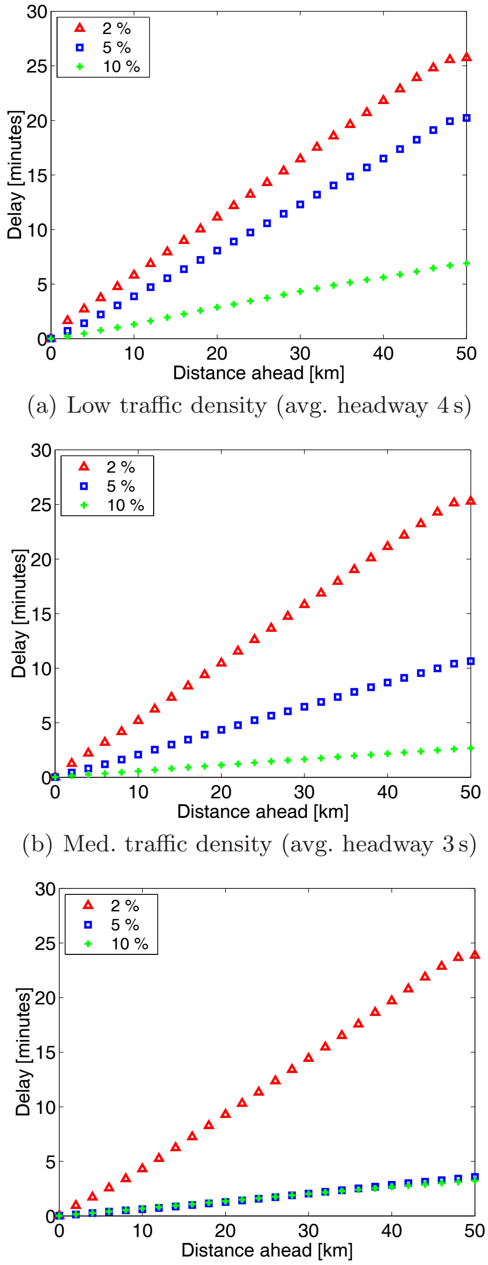
The SOTIS simulation covers the complete road traffic under realistic conditions. It is assumed in different simulation runs that 2-10 % of all cars are equipped with SOTIS. A road segment has a fixed length of 500 m. Each SOTIS equipped vehicle transmits 2 broadcast packets per second. Of these packets, 66 % include traffic information for the road that the vehicle travels on, the remaining packets are used for transmitting information for other roads. For evaluating the system performance, the information stored in the knowledge base of each car is analyzed every 500 ms of simulation time and the respective delay with which each information element has been received is calculated.

8.5.3 Simulation Results

The main objective of the simulations was to evaluate if the SOTIS technology can provide traffic information with a reasonable delay. One important result is shown in Figure 6(b), where results from a typical scenario with an average headway of 3 s are shown. The average delay of information about a specific road segment increases linearly with its distance to the current position. For a situation where 10 % of all vehicles are equipped with SOTIS technology, the information delay is extremely small ($\approx$ 3.5 s/km). Even if only 2 % of all vehicles are equipped, information is successfully distributed. In this case, the delay increases to $\approx$ 31.4 s/km, which is still acceptable for such applications, given the relatively large time scale on which traffic conditions usually change. Note, that the traffic analysis still can be *displayed instantly* to the driver, since the vehicle continuously updates its knowledge base.

Figure 6(a) and Figure 6(c) illustrate the effect of relatively low and high traffic density on the information delay. In case of a low traffic density, the delays for a market penetration of 5 % and 10 % increase significantly. This is due to the fact that the chance for a vehicle in transmission range decreases, therefore the information is less often forwarded using the air interface – the share of transport onboard vehicles increases. For very low penetration, this way of transport is already dominating, the delay increases only slightly. A similar effect occurs for high traffic densities: In this case, the delay for scenarios assuming a high SOTIS penetration is mainly caused by the time between two consecutive broadcast transmissions, thus it is only slightly reduced with a higher traffic density.[9] However, the information delay can be reduced much further, if an adaptive method for data broadcast is applied, as proposed in Section 8.6.

[9] Another contrary effect in case of high traffic densities is that the average velocity of a vehicle decreases (by approx. 10 % compared to a low density for the simulated scenarios).



(a) Low traffic density (avg. headway 4 s)

(b) Med. traffic density (avg. headway 3 s)

(c) High traffic density (avg. headway 2 s)

Fig. 8.6. Performance of the basic SOTIS system for a penetration of 2 %, 5 % and 10 % when the traffic density is varied.

The performance results in these road scenarios demonstrate the potential of the SOTIS technique. The system is able to provide information for the local area of the vehicle with reasonable delay even if only a low penetration of 2-3 % is assumed. Until now, the considered basic SOTIS implementation was assumed to be static: Broadcast messages were generated at constant intervals and the transmission range/transmission power of the vehicle was assumed to be fixed. Overload conditions were not actively avoided – their effect was simply mitigated by the high level of redundancy due to the periodic repetition of the broadcast messages. In the next section, this basic system is extended with a heuristic approach for actively avoiding overload conditions and favoring the propagation of significant changes.

8.6 Adaptive Broadcast

Since the SODAD method and its application in SOTIS are based on the distribution of information using recurring (1-hop) broadcasts, the duration between two subsequent transmissions is a crucial parameter. On the one hand, a short inter-transmission interval can help to reduce the time required to distribute new information within the system, but results in a large amount of required transmission bandwidth per vehicle and will lead to more packet collisions. On the other hand, choosing a larger interval reduces the bandwidth requirements but increases the delay and the risk of missing communication opportunities (e.g. a vehicle passing by at a high relative velocity).

Therefore, the focus of this section is the dynamic adaptation of the broadcast interval in order to distribute (traffic) information efficiently. A suitable adaptation scheme called *Provoked Broadcast* is presented and afterwards compared with the regular, strictly periodic broadcast, which was assumed previously.

8.6.1 Challenges and Requirements

Adaptive information dissemination in the considered vehicular ad-hoc network is a challenging task: The environment is highly dynamic (e.g. on highways relative velocities of up to 400 km/h can occur) and the density of vehicles can vary from 1-2 vehicles per kilometer in low density night traffic to more than 500 nodes/km in traffic jam situations. Additionally, depending on the transmission range of the air interface, these node densities can change completely within times in the order of seconds – for example if a road with very low traffic density intersects with a crowded highway.

Whereas in low density situations a large transmission range is advantageous, in high density situations it leads to a decrease of the available transmission bandwidth for an individual vehicle. Analogously, in low density situations a short inter-transmission interval is beneficial, but it can lead to overload conditions in situations of high density. Basically, the following methods could be used to solve this problem:

i) variation of the transmission range (power control)
ii) variation of the inter-transmission interval, and
iii) combined approaches.

In this chapter, the focus is on the variation of the inter-transmission interval for the following reason: It is assumed that the transmission range of the air interface is much smaller than the area that an individual vehicle is interested in. For example, the air interface proposed in [10] has a transmission range of approximately 1 km, whereas in SOTIS each driver is interested in information on segments within 50-100 km of the current position. Therefore, it is assumed that reducing the transmission range in high density situations is unlikely to reduce the bandwidth needed for information distribution significantly, since under these conditions decreasing the transmission range increases the number of transmissions necessary for forwarding the information.

The performance of a broadcast scheme can be characterized by the combination of two properties: required bandwidth (mean) and average deviation of information available in a vehicle compared to the actual value of a segment. This average deviation (i.e. mean error) will usually depend on the distance of a segment. Additionally, the scheme should be as simple as possible and has to be able to adapt to the rapidly changing conditions in IVC.

8.6.2 Adaptation Procedure

As outlined before, a strictly periodic broadcast (constant inter-transmission interval) can lead to a high number of collisions – especially in situations with high node densities. In contrast, the heuristic approach for the adaptation of the transmission interval, called *Provoked Broadcast* in the following, adapts the inter-transmission interval to the local environment and knowledge gained from received packets in order to

– reduce the delay with which information is propagated,
– favor the propagation of significant changes,
– avoid redundant transmissions, and
– occupy less bandwidth in cases of congestion.

The basic idea is the following: A default inter-transmission interval T_{upd} is chosen small enough to recognize a vehicle passing by at the maximum relative velocity. If a maximum relative velocity of 500 km/h and a transmission range of 1000 m is assumed, an interval of 5 s is sufficient to recognize and inform any vehicle. This default interval is adapted according to two kinds of observed events:

1. **Provocation:** A *provocation* is an observed event that reduces the time that elapses until the next broadcast packet is transmitted.

2. **Mollification:** A *mollification* is an observed event that increases the time that elapses until the next broadcast packet is transmitted.

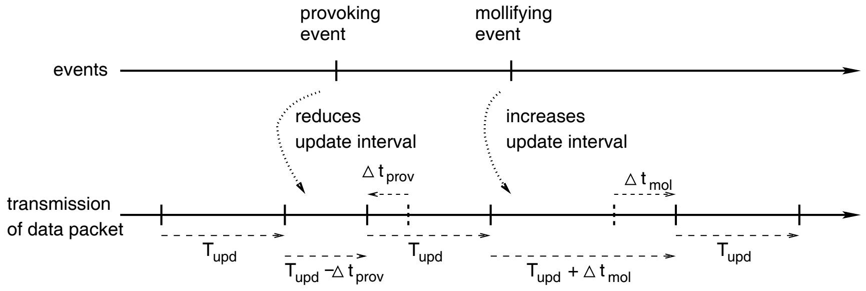


Fig. 8.7. Adaptation of the inter-transmission time in the *Provoked Broadcast* scheme.

Table 8.2. Examples for provoking and mollifying events in the Provoked Broadcast scheme.

Examples for Provocations

Event	**Intention**
Reception of information being out-of-date	Transmitting vehicle needs updated information
Reception of packet with significantly different new information	Favor propagation of changes
Reception of information from vehicle with large distance	Favor large hops in propagation
Indication (e.g. by lower layers) of excessive bandwidth	Decrease delay of information propagation

Examples for Mollifications

Event	**Intention**
Reception of similar/more up-to-date information from nearby	Avoid redundant transmissions
Indication that number of received reports exceeds threshold	Limit maximum used bandwidth

The scheme is illustrated in Figure 8.7, where the default inter-transmission interval T_{upd} is decreased by Δt_{prov} when a provoking event occurs, and is increased by Δt_{mol} when a mollifying event occurs. Herein, Δt_{prov} as well as Δt_{mol} can either be determined relative to the remaining time until next transmission of a traffic report or be chosen as an absolute value. Examples for provoking and mollifying events, which will also be used for simulations (Section 8.6.8), are listed in Table 8.2.

8.6.3 Parameters of Adaptive Broadcast

Upon the reception of a data packet, its content is examined in order to update the vehicle's knowledge base and to determine if a provoking or mollifying event has occurred. A received data packet is composed of the information for multiple (usually in the range of several hundred) road segments. Each received per-segment information value is compared to the data currently available in the local knowledge base. If the received information is assumed to be more accurate (determined e.g. by the time-stamp), the knowledge base is updated accordingly.

Additionally, based on the comparison of the received data and its time-stamp with in the knowledge base for each individual road segment, a weight $w_{m,n}$ of a received message m at node n is calculated: It indicates the difference of the received per-segment data compared to the node's previous knowledge. The decision if an information value is significantly newer or different than the previously available information is based on two threshold values: If the difference of the two time-stamps exceeds the threshold ΔT_{th}, $w_{m,n}$ is increased by a constant q_{date} (the so-called date quantum). Analogously, if the difference of the two information values exceeds the threshold ΔI_{th}, $w_{m,n}$ is increased by q_{info}.

Thus, the weight of a received message composed of information values for S distinct segments[10] of a road is calculated as

$$w_{m,n} = \sum_{i=0}^{S-1} w_{\text{info}}(s_{m,i}, s_{n,i}) + w_{\text{date}}(t_{m,i}, t_{n,i})$$

where

$$w_{\text{info}}(s_{m,i}, s_{n,i}) = \begin{cases} q_{\text{info}} & : \quad |s_{m,i} - s_{n,i}| \geq \Delta I_{\text{th}} \\ 0 & : \quad |s_{m,i} - s_{n,i}| < \Delta I_{\text{th}} \end{cases}$$

$$w_{\text{date}}(t_{m,i}, t_{n,i}) = \begin{cases} q_{\text{date}} & : \quad |t_{m,i} - t_{n,i}| \geq \Delta T_{\text{th}} \\ 0 & : \quad |t_{m,i} - t_{n,i}| < \Delta T_{\text{th}} \end{cases} \tag{8.2}$$

Here, the values $t_{m,i}$ indicate the time-stamp for segment i in message m and $t_{n,i}$ the time-stamp for segment i in the knowledge base of node n. Analogously, $s_{m,i}$ and $s_{n,i}$ are the respective information values for segment i.

By this threshold mechanism, a message received from a node which has significantly different information for the segments included, will be assigned a high weight $w_{m,n}$. In contrast to this, a low weight means that the node that broadcasted this message has a very similar view of these segments. In the following, it is assumed that the constants q_{info} and q_{date} are chosen in

[10] The number of road segments S in a message depends on the maximum packet size of the used radio interface. Typically, S will be in the range of several hundred.

a way that $S \cdot (q_{info} + q_{date}) \leq 1$ and therefore the weight of a message is in the interval $[0, 1]$.

8.6.4 Provoking and Mollifying

Based on the weight of a received message, a node determines if a provocation or mollification has occurred: Reception of a message with a weight less than the mollification weight w_{mol} causes an increase of the remaining time by Δt_{mol} until the next transmission of a traffic analysis for the respective road segments; reception of a message with a weight larger than the provocation weight w_{prov} decreases this time by Δt_{prov}. Both values can either be chosen as absolute times or relative to the currently remaining time until the next transmission. In general, the grade in which the current interval is adapted should correspond to the weight, e.g. a high weight (≈ 1) should cause a more significant adaptation of the interval than a weight just slightly larger than w_{prov}. A detailed explanation of the parameters used in the simulations is given in Section 8.6.8.

8.6.5 Interdependence of Provoking and Mollifying

Boundary condition for the outlined self-organizing system is that the bandwidth is limited. One approach is to use the event that the desired network load is exceeded as a mollification. However, in many cases this is not necessary due to the interdependence of provoking and mollifying events outlined in the following example: Consider a cluster consisting of M individual nodes, which are all in transmission range of each other. Since they are at a similar physical location, they will have a similar view of the surrounding environment. Now, an approaching node transmits a data packet with significantly new information. The weight computed in each of the receiving nodes will therefore be high and all will reduce the remaining time until their next transmission. However, since the nodes are not synchronized, one of them will transmit first. When the $(M - 1)$ other nodes receive this data packet, a low weight is assigned (since they all have similar information) and a mollification is caused. Therefore, the provocation caused by the approaching node causes only one transmission immediately after the reception. An interesting question – which will be considered next – is, which of the M nodes in the cluster should perform this transmission.

8.6.6 Influence of Distance

In a self-organizing network based on broadcast messages, it is common sense that favoring large hops in the propagation of information can be used to reduce the required bandwidth. Furthermore, nodes at a larger distance are more likely to be out of transmission range in the near future. Therefore, the

Provoked Broadcast scheme is includes a distance threshold D_{th} that is used to favor the propagation by using large hops. A distance quantum q_{dist} is calculated, depending on the distance d_{tx} to the transmitting node

$$q_{dist}(d_{tx}) = \begin{cases} 0 & : \quad d_{tx} < D_{th} \\ \frac{d_{tx}}{d_{tx,max}} & : \quad d_{tx} \geq D_{th} \end{cases} \tag{8.3}$$

where $d_{tx,max}$ is the maximum transmission range and d_{tx} is the distance of the node that transmitted the message (calculated using position information in the packet header).

If $q_{dist} > 0$, a provocation is caused, where Δt_{prov} is set to q_{dist} times the previously remaining time until the next transmission. Therefore, more distant nodes are more likely to transmit next. Since their transmissions will cause mollifying events in nodes in transmission range, large hops are favored in the propagation of information.[11]

8.6.7 Potential Risks

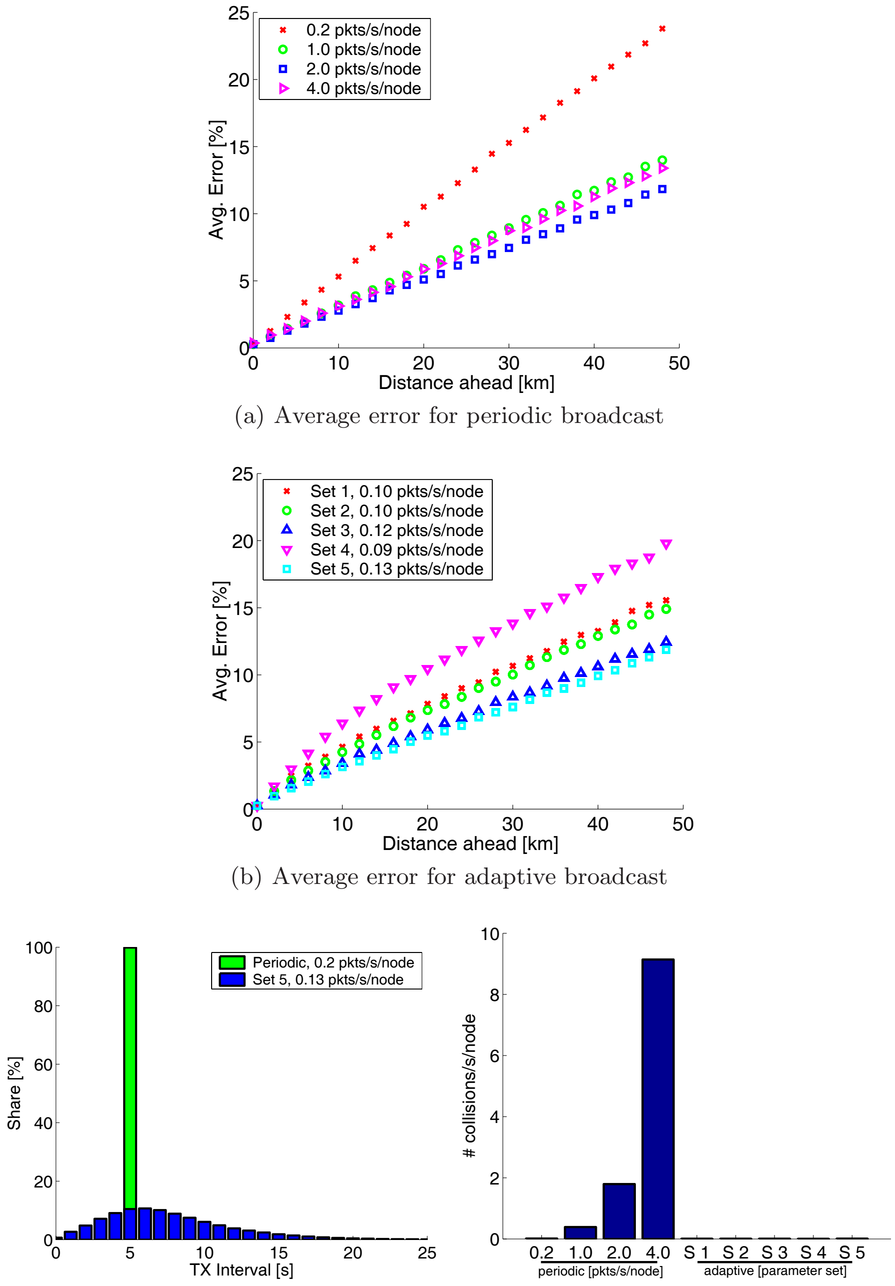
A minor disadvantage of the proposed scheme is that in cases where a strong provocation occurs (i.e. reception of a packet with $w \approx 1$) within a cluster of nodes, the nodes will collectively reduce their remaining time until the next transmission of a traffic analysis. This can increase the data packet collision rate slightly. However, a suitable MAC protocol can be used to avoid this risk – and furthermore our simulations indicate that even with standard wireless MAC protocols currently in use the rate of collisions is relatively low.

8.6.8 Performance Evaluation

Table 8.3. Parameters for the adaptive broadcast scheme.

Param.	PBcast	PBcast	PBcast	PBcast	PBcast
	Set 1	Set 2	Set 3	Set 4	Set 5
q_{info}	0.005	0.010	0.020	0.001	0.040
q_{date}	0.005	0.005	0.000	0.001	0.005
ΔT_{th}	60 s				
ΔI_{th}	10				5
ΔD_{th}	750 m				
w_{prov}	0.04				
w_{mol}	0.01				
$T_{Ichange}$	300 s				

[11] A similar approach is the basis of the recently proposed beaconless or contention-based geographic routing algorithms [11, 12].



(a) Average error for periodic broadcast

(b) Average error for adaptive broadcast

(c) Histogram of transmission intervals

(d) Packet collisions

Fig. 8.8. Performance comparison of the adaptive scheme with a periodic broadcast: average error of available information, transmission intervals and collisions.

For performance evaluation, a simulation of the vehicular ad-hoc network analogous to Section 8.5 was conducted. The same road traffic model and parameters (Table 8.1) were used, except for the following simplification to limit the simulation effort: One road of circular shape with two lanes per direction and a length of 64 km was simulated. Furthermore, for all scenarios a penetration rate of 10 % and an average headway of 3 s was assumed. As a result, the total number of vehicles simulated was reduced to about 5800 vehicles. Per scenario, 8000 s of simulation time were simulated. Of these, the last 6000 s were used for calculating the statistics.

Table 8.3 describes the different parameter sets which where used for the system parameters introduced in Section 8.6.3: The main parameters varied over the five sets are the info quantum q_{info} and the date quantum q_{date}. In combination with the threshold values ΔI_{th}, ΔT_{th} and the weights w_{prov} and w_{mol} these determine how fast the adaptation is performed.

The values for Δt_{prov} and Δt_{mol} by which the currently remaining time t until the next packet transmission is adapted (see Section 8.6.2 are calculated as follows: Δt_{mol} is constantly set to 1.0 s, Δt_{prov} is $t(1 - q_{dist})(0.5 + w)$. The optimal values for these variables as well as for the other parameters depend on the application – however, as the simulation results show, the scheme is not very sensitive to variations of these values.

The required bandwidth for the adaptive broadcast scheme depends on the data to be distributed. In order to have a worst case estimate, a uniform distribution of the per-segment info values was assumed (maximizes entropy). It was updated independently and randomly for each segment with the change period $T_{Ichange}$. In reality, the information values will probably have less entropy due to the spatial component of the distributed information and appropriate coding techniques could significantly reduce the amount of data to be transferred. These aspects are out of the scope of this chapter.

Four periodic and five adaptive scenarios have been simulated. For the periodic scenarios, the data rate for each vehicle was varied from 0.2 to 4.0 messages per second per vehicle. Figure 8(a) shows that increasing the rate from 0.2 to 1.0 pkts/s/node significantly improves the available information, whereas a further increase results in only a minor improvement. For 4.0 pkts/s/node the accuracy of the information available even decreases. The reason is that in constellations with high node densities too many packets are lost due to collisions, see Figure 8(d).

For the adaptive broadcast scheme shown in Figure 8(b), the first important observation is that the performance of the periodic scheme can be achieved with about 1/10th of the required bandwidth, e.g. comparing the adaptive scheme with parameter set 1, 0.1 pkts/s/node, to the periodic scheme with 1.0 pkts/s/node. Furthermore, even more "aggressive" parameter sets, e.g. set 5, increase the accuracy of the available information but also require some additional bandwidth. The superior performance of the adaptive scheme is explained by Figures 8(c) and 8(d): The number of collisions is reduced and the transmission interval is adapted to the current situation and available information of a vehicle. The indicated performance improvement clearly compensates the additional computational overhead introduced by the adaptive scheme.

8.7 SOTIS Prototype

Encouraged by the results of the simulations, the next step was a prototype implementation of SOTIS. The main motivation is to demonstrate the feasibility and the features of a self-organizing traffic information system – for small scenarios, the validation of simulation results is also possible. Requirements for the demonstrator were:

Demonstration and experimental experience: The prototype system completely implements the proposed SODAD method as well as its application for the dissemination of traffic information in SOTIS. The advantages of SOTIS compared to conventional traffic information systems can be demonstrated in realistic situations. Furthermore, the prototype includes monitoring functions, e.g. for analyzing the transmitted data packets and for the visualization of all vehicles in the (1-hop) transmission range.

Modularity: In order to allow a flexible prototype configuration for several demonstration scenarios, it is composed of 7 independent modules which communicate using documented interfaces. This allows us to use different types of sensors for sensing traffic information, multiple user interfaces for visualization and different mapping components.

Platform independence: The application is platform independent. Currently Windows and Linux platforms are supported but basically any OS can be used, where a Java virtual machine is available.

Off-the-shelf hardware: In order to limit the costs and development effort for this first prototype, it was decided to use off the shelf hardware, especially for the air interface.

8.7.1 Components

The functionality of SOTIS is realized by the 7 individual components illustrated in Figure 8.9. Their individual task and implementation is briefly described in the following.

SOTIS Core

The main system component, which coordinates the processing of traffic information and the composition of new data packets, is the *SOTIS Core*. Data is acquired using the three ”lower” interfaces to Sensor, Position and Communication components, processed and stored, transmitted or displayed to the user. The selection of the information to be included in data packet created by a vehicle is based on the importance values calculated in the knowledge base.

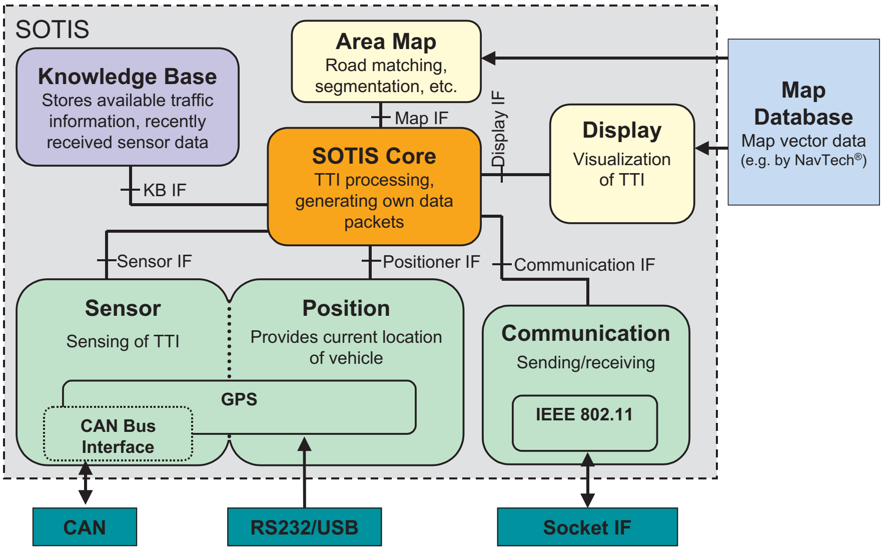


Fig. 8.9. Block diagram of the SOTIS prototype.

Knowledge Base

The *Knowledge Base* stores the per-segment information available in each vehicle, indexed by the road identifier. It periodically evaluates the importance of each segment and discards information of low relevance, e.g. if the segment time stamp is outdated or a specified memory limit is reached. The knowledge base also tracks the last time that a segment has been transmitted.

Area Map

The main functions offered by the *Area Map* component are map matching and segmentation. Geographical coordinates can be converted to the triple of (road identifier, segment number, direction) and vice versa. Map information can either be obtained from a commercial vector map[12] or from a proprietary map format, which is based on a combination of scanned bitmap images and vector data.

Display

Visualization of the currently available information, the local area of the vehicle and all vehicles in direct communication range is handled by the *Display* component. In a product implementation, these tasks would be part of the navigation system.

[12] Currently, access of *NavTech*® maps using a proprietary interface provided by *Mapsolute*® is supported.

Position

The *Position* component provides information on the current location of the vehicle. Currently, three variants of this component exist: The position can be obtained from a previously recorded trace file, via TCP/IP from a network socket or using a GPS receiver. The latter one reads GPS data conforming to the NMEA-0183 standard from a RS232 or USB port and has been tested with various commercial GPS receivers.

Sensor

Similar to the previous component is the *Sensor*. It determines the traffic information for the current location of the vehicle. For the prototype, information on the average velocity of a vehicle in a segment is used, which can be obtained via GPS, from a trace file/network socket or from the CAN bus of the vehicle. Due to the similarity of the *Positioner* and the *Sensor*, these interfaces are sometimes implemented by one module, e.g. the adaptor for parsing GPS information.

Communication

A basic requirement for inter-vehicle communication is a suitable wireless air interface. Although it has been shown that the IEEE 802.11 standard has severe problems in vehicular scenarios [9], it is used in the SOTIS prototype simply because no other more suitable transceiver was available at a comparable cost. In order to avoid problems with external antennas, USB variants of commercial IEEE 802.11b wireless LAN cards were used (DSSS, 1 Mbit/s mode). These were installed at the roof of the vehicle and connected via USB. Since all communication related functions are capsuled in the *Communication* component, any other wireless network interface could easily be used. Two alternative modes of communication were implemented: communication via standard UDP/IP and via the FleetNet router.

8.7.2 Test Results

Three test vehicles have been equipped with the proposed system. Typical scenarios have been successfully tested: Gathering traffic information in each vehicle, information dissemination in a line of vehicles and information dissemination onboard a vehicle on the opposite lane. The system has been tested in various parts of northern and central Germany.

The prototype application is also able to record trace files of positions, sensor information and transmitted/received data packets. Using the collected trace files and a wireless network emulation which artificially limits the transmission distance of a vehicle based on its geographic position, critical situations can also be reproduced and analyzed under laboratory conditions.

8.8 Conclusion

Inter-vehicle communication can significantly increase passenger safety and comfort. However, the radio channel and network characteristics in the vehicular environment are technically complicated and a real challenge for designing suitable air interfaces, protocols and applications. A critical requirement for market introduction is that the system offers significant benefit even if only a low ratio of all vehicles is equipped.

The main contributions of this chapter are the following: first, a novel method for data abstraction and dissemination in vehicular ad hoc networks is proposed. It is targeted at the distribution of data with a spatial component in sparsely connected mobile ad hoc networks. Second, this method is applied in a Self-Organizing Traffic Information System (SOTIS) for the distribution of detailed and up-to-date travel and traffic information for the local environment of a vehicle. Performance evaluation by means of simulation shows that SOTIS requires only a very low penetration ($\approx 1-3\,\%$) and outperforms conventional centralized systems. Third, an experimental SOTIS prototype is presented. Based on off-the-shelf hardware, it demonstrates the feasibility of the proposed system and allows the acquisition of experimental data.

Acknowledgement

The authors would like to thank the companies *NAVTEQ*® and *Mapsolute*® for providing the digital map data.

References

1. Wischhof, L., Ebner, A., Rohling, H., Lott, M., Halfmann, R.: SOTIS – a self-organizing traffic information system. In: Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, Jeju, Korea (2003)
2. Michael, L.B., Nakagawa, M.: Non-platoon inter-vehicle communication using multiple hops. IEICE Trans. Commun. **E82-B** (1999) 1651–1658
3. Michael, L.B.: Adaptive layered data structure for inter-vehicle communication in ad-hoc networks. In: ITS 2001, 8th World Congress on Intelligent Transportation Systems, Sydney, Australia (2001)
4. Briesemeister, L., Hommel, G.: Integrating simple yet robust protocol layers for wireless ad hoc intervehicle communications. In: Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS). (2002) 186–192
5. Dashtinezhad, S., Nadeem, T., Dorohonceanu, B., Borcea, C., Kang, P., Iftode, L.: Trafficview: A driver assistant device for traffic monitoring based on car-to-car communication. In: Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC Spring 2004), Milan, Italy (2004)

6. Miller, R., Huang, Q.: An adaptive peer-to-peer collision warning system. In: Proceedings of the IEEE Semiannual Vehicular Technology Conference (VTC Spring 2002), Birmingham, Alabama, USA (2002)
7. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. I (France) **1992** (1992) 2221–2229
8. Nagel, K., Wolf, D.E., Wagner, P., Simon, P.: Two-lane traffic rules for cellular automata: A systematic approach. Technical Report LA-UR 97-4706, Los Alamos National Laboratory (1997)
9. Ebner, A., Rohling, H., Wischhof, L., Lott, M., Halfmann, R.: Performance of UTRA TDD ad hoc and IEEE 802.11b in vehicular environments. In: Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference, Jeju, Korea (2003)
10. Ebner, A., Rohling, H., Halfmann, R., Lott, M.: Synchronization in ad hoc networks based on UTRA TDD. Proc. PIMRC 2002, Lisbon, Portugal (2002)
11. Heissenbüttel, M., Braun, T.: BLR: beacon-less routing algorithm for mobile ad-hoc networks. Elsevier's Computer Communications Journal (2003)
12. Füßler, H., Widmer, J., Käsemann, M., Mauve, M., Hartenstein, H.: Contention-based forwarding for mobile ad-hoc networks. Elsevier's Ad-Hoc Networks **1** (2003) 351–369

9

The Pinboard Application

Location Based Messaging for Vehicular Communications

Jerry Cheambe, Jean-Jacques Tchouto and Carsten Tittel

Fraunhofer Institute for Open Communication Systems, Berlin, Germany
cheambe@fokus.fraunhofer.de, tchouto@fokus.fraunhofer.de,
tittel@fokus.fraunhofer.de

Summary. A Vehicular Ad-hoc Network (VANET) is a highly dynamic network that typically relies on geographical routing. It represents a migration from hot-spot communication to multi-hop ad-hoc communication. The Location-Based PinBoard Application (LBPA) described below presents a new approach to distribute short messages in predefined regions using geographical broadcast and a store-and-forward mechanism. This chapter introduces the realm of ad-hoc inter-vehicle applications, their main categories, the technical challenges and economical pitfalls. Next, the application itself, its concept, components and scenarios will be explained in detail. In addition, results from field trials are presented, followed by the lessons learned and a look into the future of applications for VANETs.

9.1 Introduction and Motivation

Users around the world increasingly integrate computers and the Internet in their lives. The deployment and usage of VANETs is one of the next natural steps in this direction. There is a slow but steady shift in the role of vehicles in the lives of consumers. Vehicles are no longer viewed only as a means of transport. Instead, consumers are beginning to expect their vehicles to provide them with information and communication services. Vehicles are viewed as extensions of the home and office environment. Passengers are increasingly interested in personalized, up-to-date location-specific information, e.g. traffic information from their direct vicinity.

Three main factors are responsible for the above trend:

1. The increase in consumer acceptance of pervasive computing and the growing need for individual-centric services.
2. The maturing of the wireless technology resulting in advanced equipment available at low costs.
3. The change in the role of vehicles in communication. Vehicles are increasingly playing the roles of sensors and relays in the communication

landscape. Furthermore, vehicles are beginning to act as servers and information sources.

Vehicle manufacturers, having recognized this trend strive to satisfy their customers while creating a competitive advantage by integrating state of the art technology into their products. As a result, more and more vehicles are equipped with wireless devices facilitating communication and navigation. These devices use short-range radio hardware and enable inter-vehicular communication based on multi-hop networks, thus providing connectivity to other networks and the Internet. Hence, communication scenarios will include vehicle-to-vehicle, vehicle-to-Internet and vehicle-to-hotspot communication.

The success of the inter-vehicular network will depend on available services. During the early stages, suitable deployment applications need to accommodate a smooth transition by coupling accepted, successful services with new safety and convenience applications. The Location-Based PinBoard Application (LBPA), which combines messaging services and location-based services is a suitable deployment application for VANETs.

The prime motivation for the development of the application was derived from the need to exchange location dependent, personal information. Based on the well known, successful Short Message Service (SMS), the PinBoard application enhances the concept of real world message boards by delivering messages to target regions, using geographical multicast and store-and-forward mechanisms. Multiple access options shall simplify the usage of the application: the PinboardMessages can either be created like short messages from mobile clients or via a web interface from any place connected to the Internet. The network not only delivers the messages to appropriate receivers, but stores and manages them for the period of validity.

Important for the development of the LBPA was also the desire to have an application that works at very low levels of market penetration, only 2% should be sufficient. Such applications provide value-added services to users in the initial stages of the technology distribution and motivate users to buy them. Hence they are called deployment applications. The PinBoard application was designed, implemented and tested by Fraunhofer FOKUS within the framework of the FleetNet project in co-operation with industrial and educational partners.

This chapter is organized as follows: Section 9.2 gives an overview of the fundamentals of ad-hoc inter-vehicle applications. In Section 9.3, the application components and architecture are described. Section 9.4 presents the implementation, explains the trials and concludes with open issues and further work.

9.2 Fundamentals of Ad-hoc Inter-Vehicular Application

Before delving into the detailed description of the PinBoard (LBPA) application, we first introduce some concepts fundamental to understanding Inter-

Vehicular Applications (IVAs): The first subsection classifies IVAs. This classification is followed by the challenges faced when designing and developing IVAs. The last section looks at some factors necessary for a quick market penetration of this technology and its long term deployment.

9.2.1 Categorization of Ad-hoc Applications

IVAs differ from one another in many aspects which not only affect the hardware environment in which they are deployed, but also the user context and the communication subsystem. Possible criteria for categorizing these applications include the level of interaction, the bandwidth requirements, data propagation, the application control architecture, quality of service requirements, context awareness and the area of use. These categories will be explained below in detail.

Level of user interaction

Applications require varying levels of user attention and interaction. An application may be interactive, informative or non-interactive. Interactive applications support bi-directional information exchange via appropriate Vehicle User Interfaces (VUIs). The passenger informs the application about his intention, either via speech, touch or gesture. The application feeds back the requested information, minimizing the distraction of the driver. Informative applications support only a uni-directional distribution of information, normally from the car to the driver, whereas non-interactive applications run in the background and act on behalf of the driver. To give some examples:

- User initiated applications will always be interactive, e.g. route planning or message creation or web browsing;
- Informative applications mostly stream information, e.g. a news cast or warning sensors; and
- Non-interactive applications may be concerned with vehicle maintenance.

Bandwidth requirements

Ad-hoc networks and VANETs in particular have very limited bandwidth. Application developers must seek to optimize the use of the available bandwidth. Applications can be classified into two groups according to the bandwidth they use: low and high. A quantitative qualification of these classes is inexpedient as absolute values are dependent on the underlying technology (WLAN, Bluetooth^{TM}etc). From a qualitative point of view, applications that support audio and video have high bandwidth requirements, while applications with sporadic data transmissions, e.g. message services, require low bandwidth.

Data propagation

Data propagation refers to the mechanisms used by the underlying network to transfer information from one node to the next to enable the proper functioning of the application. Data propagation via *routing* and via *broadcast* are identified. In the latter, data packets are broadcasted to all nodes, which then check the contents of the data packets for relevance and discard them if irrelevant. If multi-hop transmissions are supported, each receiving node may re-broadcast the packets. Data propagation via routing mechanisms transmits data packets to a predefined node or group of nodes. The routing mechanisms can be address based or location-based. While address based routing ensures that only vehicles with certain addresses would receive the packets, location-based routing ensures that only vehicles in a given geographic region can receive the data packets. All other nodes drop the packets or act as forwarders. A typical example for broadcast propagation is news cast. Address based routing is used by peer-to-peer applications like web browsing and location based routing shall be used for services like accident warning or advertising.

Application control architecture

IVAs can be characterized according to their control architecture. Principally, a centralized and a decentralized architecture can be distinguished. In the first case, there is a central instance, usually with special functionality and superior resources (a single server or a conglomeration of servers), that centrally controls the entire application across the network. Client nodes send requests to the central instance which processes them and responds to the clients. Centralization implies that at least the server has global knowledge of the state of the application.

In contrast to centralized application control architecture, decentralized applications do not demand for central server or groups of servers that have special functionality or play a coordinating role. All the functionality is usually distributed among all network nodes. In decentralized inter-vehicular networks, it is necessary to distinguish between applications that have network-wide knowledge of events from those that possess only local knowledge. Applications with network-wide knowledge propagate information on changes to all nodes in the network. This increases network traffic, but may be necessary for the proper functioning of certain applications. Applications requiring localized knowledge offer full functionality to nodes by aggregating and processing information only from the immediate neighborhood of the target nodes.

Characteristic examples of application with a centralized architecture are E-Bay or Amazon. Peer-to-peer networks or the World Wide Web itself are good examples for the decentralized ones.

QoS requirements

Quality of service refers to the level of performance that can be provided by the network. When the level of performance is unpredictable and unreliable, the service is termed best-effort service. Predictable, reliable, and guaranteed services are described as priority services.

Most computer networks today provide best-effort services, where the network performance varies with time, depending on the state of the network at a given interval. While best-effort performance is acceptable to many applications and users, it is unacceptable for certain application classes. These classes of applications require consistent, deterministic, and/or guaranteed network performance, e.g. priority access as for active safety applications and emergency services.

Quality of service can be characterized in terms of bandwidth, delay, and reliability. These and other QoS characteristics are used to define services in the network, as well as to develop service metrics that are used to verify services within the network.

Bandwidth requirements by applications cover a wide spectrum of possible combinations. While some applications, like IP telephony, require a Constant Bit Rate (CBR), e.g., due to the same sample rate and sample resolution at all time, other applications require varying bandwidth according to the current content processed. This demands are communicated by specifying different data rates: the Peak Data Rate (PDR), defining the maximum on resources needed during bursts, the Sustained Data Rate (SDR), describing an upper limit of the average rate 'sustained' during communication, and the Minimum Data Rate (MDR), giving the absolute minimum to keep the application running.

Delay sensitivity refers to the reliance of an application on timely accurate information. Two main characteristics can be distinguished: Sensitivity to end-to-end (or round-trip) delay, or sensitivity to delay variation (delay jitter). First, delay jitter sensitivity is an important factor for all applications receiving (and processing) events constantly. Second, end-to-end delay sensitivity can be classified as real-time and delay-tolerant ones. Most data transfer applications, like news cast, streaming or file download are sensitive to data corruption but can cope with huge delays, whereas sensor processing applications require delivery of information within microseconds ranges. Because accidents and, ultimately, life itself is at stake in car environments, especially safety and emergency applications have tight demands on delay conditions and must be granted higher priority to the transmission channels.

Reliability covers all aspects concerning network failures. Most important for application are factors like Bit Error Rate (BER) or packet loss rate. Applications can be grouped in error-prone and error-tolerant ones. Of course, all applications and protocols will be error-prone, if the error rate provided by the network approaches large numbers, but some application are quite vulnerable to low error rates, like unbuffered audio.

Context awareness

Context refers to any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Context aware applications utilize the context at any given time to enhance services. IVAs can thus be categorized according to the context they use, common context types are location, (user-)identity, activity and time. Because of navigation systems, the location context is often available in vehicle environments and will be used for service enhancements.

Area of Use

IVAs can be further grouped according to their area of use. Although most services are location dependent, marketable services do not come as a direct consequence of the ability to identify someone's location through a mobile device, but rather through combining location identification with additional data to provide added value to the user. The following areas of use can be identified and will be explained below: mobile commerce applications, informative and entertainment applications, driver assistance, emergency and car-centric applications.

Mobile Commerce applications:

As content delivery over VANETs becomes faster, more secure, and scalable, mobile commerce will become a method of choice for digital commerce transactions. The current spectrum of applications includes advertising along the road, remote ordering, automated billing and proximity targeting advertisement. Mobile advertising is very interesting both for user and advertisers. On one hand, the extensive use of context awareness provides unique opportunities for personalization, thus offering new opportunities to advertisers to place effective and efficient promotions on mobile environments. On the other side, the user can tailor advertising messages to contain only desired information or might accept advertising in exchange for subscription-free enhanced services.

Informative applications:

Information services refer to digitally distributed content based on location, time and user behavior. They include tourist services such as guided tours, travel guides, transportation services, parking guides and other services that can be provided to travelers moving around in a foreign city. Furthermore, such application can act as mobile yellow pages, providing a mobile user, upon request, with knowledge regarding nearby facilities. Other classes of applications are infotainment services such as information about local events and location-specific multimedia content.

Entertainment applications:

Entertainment services belong to the class of services that do not ease driving or improve driving safety, but serve the primary purpose of giving pleasure to the passengers. Chat between passengers of different cars, interactive games and audio/visual services like video on demand all fall in this group.

Driver assistance applications:

Driver assistance applications increase the scope of perception of the driver beyond his natural capabilities and assist the driver in routine tasks in the car. These applications exploit the data exchange between vehicles and aim at improving driver safety and facilitating the task of driving. A special subgroup contains applications that actively intrude on the driver to takeover certain aspects of the car in situations of eminent danger.

Emergency applications:

Emergency applications offer the ability to locate an individual who is either unaware of his/her exact location or is not able to reveal it because of an emergency situation (accident, injury, criminal attack, or breakdown). Such applications require priority access to network resources to inform ambulance and police vehicles about the state of the passenger (if possible), the overall situation and the current location.

Car-centric applications:

Car centric applications encompass all applications that are directly or indirectly related to the maintenance and well-being of the car. The ultimate object of interest for these applications is the car itself and not its passengers. Examples include remote maintenance applications, remote diagnostics, auto/self-diagnostics, monitoring of car fleets and tracking of cars in case of theft.

9.2.2 Design Challenges

VANETs differ considerably from traditional networks because of their ad-hoc nature. The network is very dynamic, with highly mobile nodes that results in a continuously changing network topology and in unstable connections between the nodes. These differences pose unique challenges to the design and implementation of applications for inter-vehicular networks. Below is a review of these challenges.

Unstable connection

Unstable network links in VANETs are primarily the result of high node mobility coupled with radio range limitations. For example, two vehicles equipped with IEEE 802.11 WLAN radios traveling at a relative velocity of $150km/h$ have a contact time[1] of approximately $20s$ after which the connection breaks down.

The instability poses major challenges to the development of applications requiring high quality of service. Although priority access to the immediate local environment is possible, network-wide priority is difficult to guarantee. Also, acknowledgments are tricky to handle especially in a multi-hop network. Let us have a look at the example of a user in a moving vehicle requesting a paid service from a stationary server via intermediate nodes. When the server responds to the request by offering a service to the client and gets no reply, the cause is ambiguous: Did the client receive the data but could not send an acknowledgment or was it already out of range before it even received the requested information. The reverse scenario is possible for a client who sends a request to the server and gets no reply. The billing in either case becomes problematic.

Limited bandwidth

In VANETs, mobility results in a lot of network overhead. Traffic for routing/route discovery, security and self-organization reduce the real throughput of the network. Furthermore, the constantly changing topology induces rapid fluctuations in the available bandwidth than is the case with traditional wired and wireless networks.

Applications developed for such networks must be designed to adjust their payload according to available bandwidth. An inter-vehicle video application for example can reduce image size by using a different compression algorithm in circumstances of constrained bandwidth. Applications also should be developed to minimize the amount of traffic they generate.

Time synchronization

Most IVAs depend on location and time information to provide their services and it is often essential that all nodes and system components operate on the same timebase. In addition, some applications allow only for very minute time tolerance, well below $10ms$. However, because the VANET nodes are physically detached from one another or may be out of touch with one another, differences in node times are unavoidable.

Because the network cannot guarantee a network wide time, applications must provide a means of synchronizing node times through the lifetime of the

[1] Signal detection, amplifier gain adaptation, node synchronization, self-organization and data exchange are done within the contact time

application. A preferred method is making use of Global Positioning System (GPS) time which can be safely assumed to be synchronized across the globe. Another aspect worth mentioning is the support for time zones. Applications must be designed to operate across different time zones as it is inevitable that the vehicles could communicate between different time zones.

Scalability

VANETs are autonomous self-organizing networks that do not rely on external management. In addition, at the beginning of operation, their ultimate size cannot be predicted. Hence, applications must be developed so that near to none external management is necessary for their proper functioning. Moreover, the application must not exceed network resources as the number of network nodes grows.

9.2.3 Critical Success Factors

Lessons learned from the success and failures of new technologies underline the fact that technological superiority is not always a sufficient condition for innovative technologies to survive in the market. It is not uncommon for a superior technology to be rejected and replaced by an inferior one because other critical factors like perceived benefits and risks, legal issues and ease of use influence the success of a new technology. We have identified a number of critical success factors that could influence the market acceptance of IVAs: the limitations of user interfaces, privacy concerns, the need for a broader definition of context, the need for personalization and the need for deployment applications.

Personal Privacy

Location-based applications in VANETs rely on accurate positioning information. However, most users do not want their location to be tracked, tracking user locations is considered intrusive. The danger of misuse is real – a fact that increases the reluctance in accepting this technology. A granular approach to releasing location information to 3$^{\text{rd}}$ party users and applications is required when designing IVAs. For example, it should be possible for a user to forward his personal information (e.g. his location) to a trusted application, but at the same time withholding this information from all other users running that application without his explicit permission. This is appropriate when the 3$^{\text{rd}}$ party is not interested in the context itself, but in the resources and circumstances associated with the location or context [1].

Another threat to personal privacy is posed by the increase in ubiquitous computing. There is a trend in extending the home environment into the car and in allowing the user to access the same services and applications using multiple devices, while at the same time making sure that the user has the same profile. Additionally, it is not uncommon for users to share devices. Computers and applications are becoming more personalized by recording user preferences and intermediating on the user's behalf. By gathering small bits of information on users, it is now possible to build a complete profile of the user. People are increasingly concerned that information gathered about them can and, time given, may be misused.

Context beyond location

Dey and Abowd [2] define context as "*... any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*" The primary context types location, identity, activity and time are more common. VANETs and IVAs underline the importance of location awareness. Spriesterbachet et al. [3] split location awareness into physical location (the position within a reference system) and semantic location (the position within the immediate environment), while Schmidt et al. [4] also identify spatial awareness (location change of an entity) as another dimension of location awareness. A combination of all three dimensions is necessary to fully qualify location-awareness. While IVAs take location-awareness into consideration, they generally tend to neglect the other aspects of context.

Application developers need to incorporate the user's identity and his style of using the application or device to be able to provide individual-tailored services. Because of the different requirements on the drivers and passengers attentions, IVAs must observe the activity, e.g., driving or waiting. Apart from identity, time and activity, other context types must be considered, e.g., infrastructure context.

Infrastructure context refers to the relationship between the application and factors like network bandwidth, reliability and display resolution. The issues of importance include the ability to adapt to infrastructure changes, e.g., bandwidth decrease or alternating technology in heterogeneous networks, to maintain a high service quality.

Limitations of Vehicle User Interfaces

Developing interfaces that are suited for the in-car environment poses a significant challenge to the manufacturers and application programmers. Interface design aspects like information visualization and audio commands are challenging because of technological difficulties, the unpredictability of the complex traffic situation and the fact that, in the worst case, life is at stake. While the interface should give the user control of the application by enabling direct manipulation and interactivity (feedback and dialog), it must not distract the driver. The following guidelines for the design of IVA interfaces were proposed [5]:

1. *Design for safety:* Drivers should be able to read displays at a glance and complete tasks in two glances. The application interface design should emphasize simplicity, thereby increasing clarity.
2. *Design to reduce complexity:* Assistance to the driver should take priority over powerful features. The use of defaults should be encouraged for the most frequent tasks. The applications should also allow customization of information via shortcuts. Also, physical controls should be used for a single function rather than a combination of functions.
3. *Sparing use of Graphical User Interfaces (GUIs):* GUIs are less direct and require more driver attention. They should be avoided for simple tasks.
4. *Reducing cognitive overload:* The driver's focus is on the road. Applications must consider that his attention is a scarce resource. Applications should, where possible, have different versions for teenagers and the elderly.

IVAs must support intuitive input and output methods like speech recognition using a combination of both dialogue-based speech and haptic touch screens. Though these technologies are the best on the market now, they are not yet mature and ready for the mass market.

Experience from the use and adoption of mobile hand-held devices has shown that users tend to pardon the physical limitations of the devices, but are very critical of flaws in the logic of the interface [6]. Hence while hardware limitations may be tolerated, interface design errors are critical. Market success of inter-vehicle applications is dependent on how easy they are to master. Before the drivers will accept the new technology, it must become proficient almost immediately or they would be discouraged to use it.

The need for personalization

The need to accommodate for social and demographic factors like age, branding, cultural diversity and emotional appeal press the case for the personalization of inter-vehicle applications. This need is well exemplified in the mobile phone markets where look and feel of the hardware is used to differentiate and accommodate a certain degree of personalization.

The need for deployment applications

Widespread use and acceptance of IVAs presupposes that the required equipment is installed in vehicles. Presently, car and communication device manufactures are undertaking huge efforts to enhance the comfort and safety of driving. Notably, in research Inter-Vehicular Communication (IVC) technology for 'active-safety' has become a main topic in recent years. As the name implies, these systems constantly monitor the immediate environment of the car and actively communicate to increase the sensor range and avoid dangerous situations if identified. These enhancements work only in a high market penetration. To reach the needed penetration rates, initial buyers need to be convinced that the devices and applications are worthwhile, that the benefit from investing into this technology is not reached in the dim future but immanent from the beginning. However, as stated before, active-safety applications alone cannot achieve this instant conviction.

A solution to the problem is that the Original Equipment Manufacturer (OEM) can subsidize the market introduction of the equipment by offering it free to the users until a threshold penetration rate is achieved. This is not very feasible in the present financial situation and taking the high research and development costs into account. A second solution is to offer the user other value added services that work even at small market penetration rates. Without appropriate deployment applications that can offer consumers benefits at low penetration, IVAs may not attain the critical mass required to succeed as a market dominant technology. The LBPA described in the next section is an example of a deployment application.

The previous section described the fundamentals of IVAs (Inter-Vehicular Applications). First, a comprehensive taxonomy of such applications is offered, including the level of user interaction, bandwidth requirements, propagation mechanisms, application control architecture, quality of service, and the area of use. The section further identifies the challenges in designing an IVA posed by the network, the potential users and the technology. The considerations are wrapped up by discussing the factors necessary for the proper functioning, acceptance and ultimate success of inter-vehicle applications.

9.3 PinBoard Application

After introducing the fundamentals of IVAs in the previous section, this next section sets out to describe in detail an example of such an application. The application in question is a location-based messaging service known as the Location-Based PinBoard Application (LBPA). First, an overview will be given and an effort will be made to classify the application using the criteria discussed above. Thereafter, a detailed description of the architecture will be given.

9.3.1 Overview

The PinBoard application models the concept of real world notice boards: A message is pasted on the notice board that is located in the region of relevance, targeted readers pass by and read the messages that have interesting captions. Translating this idea into the world of IVC, we developed an interactive broadcast application that runs on the FleetNet VANET. The LBPA is used to distribute short messages, i.e., PinboardMessages, in predefined regions using geographical broadcast and store-and-forward mechanisms. Vehicle or Internet clients running the LBPA generate PinboardMessages. These messages specify an area of validity where they are distributed, a lifespan, after which the message is no longer valid, and the content itself. FleetNet Gateways (FGWs) (see chapter on 'Internet Integration') in and around the addressed region periodically broadcast the message to passing vehicles during the lifetime of the message. The vehicles pick up the messages and, if the message lifespan has not elapsed, the message is stored and forwarded to neighboring vehicles. If the vehicle is currently in the area of validity of the message, the message is displayed to the passengers. Else, the message is stored and displayed later when the vehicle enters the area of validity. If the lifespan of a message expires, it is dropped by the application.

The LBPA has two main areas of application: Non-time-critical traffic information and information about places of interest, business and entertainment in the immediate vicinity. While the first concentrates on data aiding the driver by providing information about free car parks, accident, traffic jams or road works, the second one is more concerned with activities outside the car, including tourist information or advertisements for special offers.

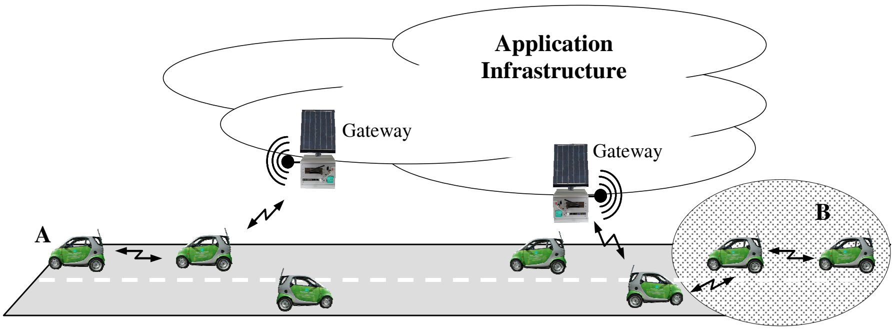


Fig. 9.1. PinBoard application – a typical use case

Figure 9.1 shows a typical use case for the LBPA. The vehicle at point A experiences an event, e.g., a traffic jam. Using a standard functionality (e.g., pushing a button), the driver generates a message describing the traffic jam. The message is targeted for area B a few kilometers behind client A. The message is sent via the VANET to the nearest FGW. A server within the application infrastructure receives the message and propagates it to appropriate gateways closest to the addressed region B. One car within the broadcast range of the gateway receives and stores the message. It then forwards it to other cars within its radio range. This process continues, increasing message reach via multi-hop communication until the message gets to the target region.

The mechanism described in this scenario is basically the same for all other scenarios. Premium services on top of the LBPA differ only in two ways. Information services like tourist information diverge from the simple scenario in their distribution nature: information is specially requested (pulled) by the clients. The second type of premium service – marketing along the road –

requires additional methods for authentication and authorization to use the commercial side of the LBPA.

After briefly describing the PinBoard application, we now move on to classify it using the criteria discussed in section 9.2 before describing the details of its architecture. The LBPA is an interactive application, requiring low bandwidth for text based messages. It is an application with centralized control in which information is propagated via routing between the server and the gateways, via broadcast between the gateways and the vehicles, and via multi-hop geographic routing between the vehicles. In its most basic scenario (the distribution of text-based messages), only best effort quality of service is necessary. As the name implies, the PinBoard application is a location-based application, adapting itself to the user's preferences and identity. It functions primarily as an informative or m-commerce application.

9.3.2 Architecture and Components

The basic architecture of the LBPA is depicted in figure 9.2.

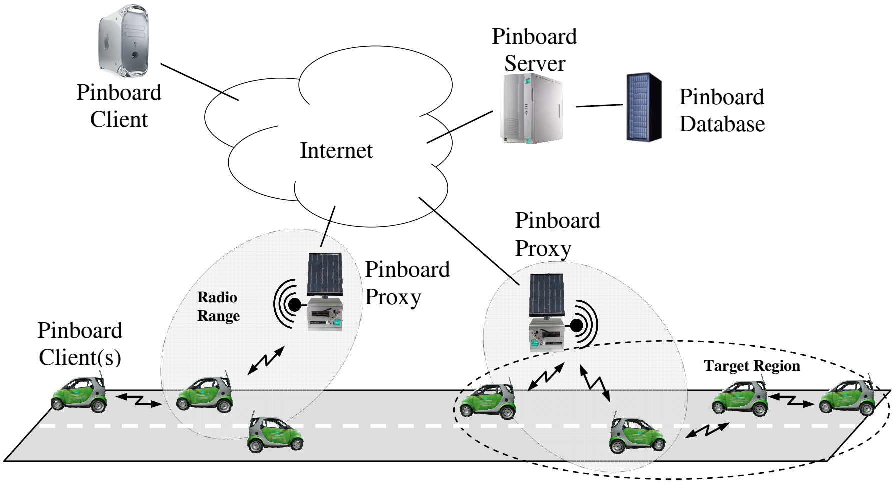


Fig. 9.2. PinBoard application – Architecture

The application consists of four core components: the PinboardClient, the PinboardServer, the PinboardProxy and the PinboardDatabase. These components are clearly defined. Their extensible interface specification makes the LBPA modular and capable of accommodating future extensions.

Application relevant regions are also highlighted in figure 9.2: The radio range of the PinboardProxies and the target area of the message. Inside the targeted area, vehicles with and without direct communication coverage can be found.

The PinboardClient

The PinboardClient runs either in a vehicle or as an Internet client. It creates and displays PinboardMessages, acting like a personal information manager application within the VANET. Normally, the PinboardClient first displays the message headers. If the user is interested, he can then request the complete message either by clicking on it or using appropriate speech commands. High priority messages like warning and emergency messages are displayed completely upon reception. PinboardMessages can be generated by the PinboardClient using either text-to-speech modules, buttons for some pre-defined standard messages or a keyboard. To transmit a message, the client needs the address of the (well known) PinboardServer. It then opens a peer-to-peer IP connection to the server and transmits the message.

There are two ways of receiving a message: the pull service and the push service.

Pull service: For the pull service, the PinboardClient creates a direct IP connection (possibly tunneled through the VANET) to the PinboardServer and demands information from the PinboardServer. The server responds with the desired content via a direct IP connection.

Push service: The push service allows the PinboardClient to receive messages via geographical broad-, multi- or anycast.

There is one fundamental difference between Internet and vehicle clients. While both are able to transmit messages and receive messages via the pull service, only vehicle clients can receive via the push service, since it is broadcast-based and runs in the FleetNet VANET.

The PinboardProxy

The PinboardProxy is responsible for receiving, evaluating, storing and forwarding messages. It runs both on stationary nodes, e.g. FGWs, and on the mobile nodes – vehicles. When configured to run on a stationary node, it acts as a PinboardProxyGateway, whereas when running on a mobile node, it is acts as a PinboardProxyClient.

The PinboardProxyClient is always attached to a PinboardClient. It acts primarily as relay for other vehicles and filters messages destined for its client. When the PinboardProxyClient receives a message, it first checks if the message is valid by verifying if the message creation time plus message lifespan is less than the current time. If the message is no longer valid, the PinboardProxyClient drops the message. Otherwise the message is forwarded to its immediate neighbors. Next, the proxy compares the target area of the message with the current vehicle location. If the vehicle is in the target area, the message is forwarded to its PinboardClient, unless the client had previously received the message. If the vehicle is not located inside the target area, the PinboardProxyClient stores the message and delivers the message to the client when the vehicle enters the target region. Messages generated by the

PinboardClient are sent to the PinboardProxyClient for transmission into the network.

The PinboardProxyGateway runs on stationary nodes. It receives messages from the PinboardServer via normal IP connections and broadcasts these toward the mobile PinboardClients.

The communication between PinboardServer and PinboardProxyGateway is enabled by a registration process: To be considered as an active stationary relay, the PinboardProxyGateway stations must register themselves, their IP addresses and locations with the PinboardServer. The registration occurs during the start-up of the PinboardProxyGateway, while de-registration occurs during it's shut-down. The PinboardProxyGateways periodically send update messages to the server informing it that they are still online. If the server does not receive an update message from a PinboardProxyGateway within a given period, the server assumes the proxy has gone offline.

The PinboardServer

The PinboardServer is able to receive, evaluate, store, transmit and retransmit PinboardMessages. The PinboardServer is also responsible for managing the PinboardProxy registration process. The PinboardProxies inform the PinboardServer when they start up (attach), when they shut down (detach) and periodically when they are running (update).

When distributing PinboardMessages, the server sends the messages to the specified addressed region. This is accomplished by selectively sending the messages only to stationary PinboardProxies that are located in or nearest to the given geographical region. The PinboardServer is also responsible for answering requests sent via the pull service, thus providing direct client-server communication.

The PinboardDatabase

The PinboardDatabase stores, manipulates and retrieves messages and proxy information. The ability to handle and index geographical data was the main criteria for the selection of the database back-end. Naturally, it can run both as distributed or centralized instance.

9.3.3 Example Scenario

Using the PinBoard application components explained above, figure 9.3 depicts a sample LBPA scenario. This scenario illustrates the propagation of PinboardMessages, from creation at an Internet client to delivery in the target region.

1. A PinboardClient creates one or more PinboardMessages and sends these messages to the PinboardServer. The PinboardClient can run on a normal

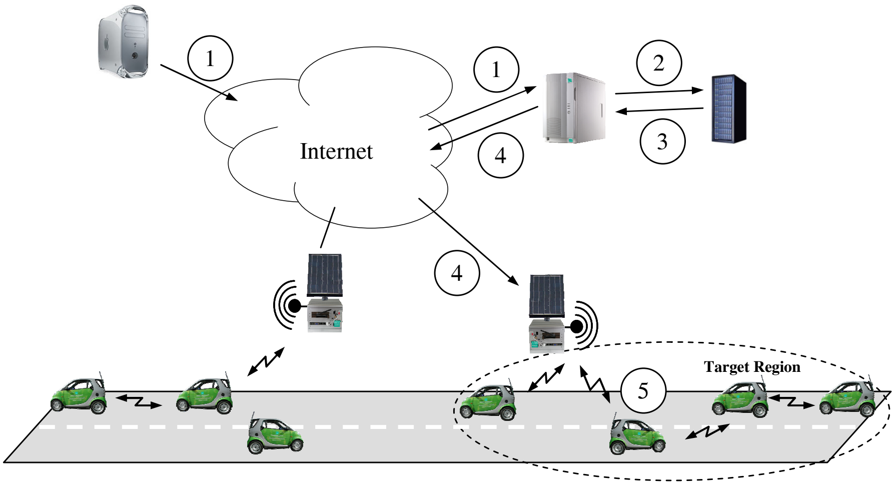


Fig. 9.3. PinBoard application – Sample scenario

node connected to the Internet or on a vehicle capable to connect to the VANET. In the latter case, the message is tunneled over the VANET to the PinboardProxyGateway, which then forwards the message to the PinboardServer.

2. The PinboardServer authenticates the message originator. It checks the message itself and saves it in the PinboardDatabase.
3. The PinboardServer periodically fetches valid messages from the PinboardDatabase.
4. The messages are send to the appropriate PinboardProxies in or near the given geographical regions.
5. The proxies then geo-broadcast all messages to vehicles within radio range.

9.4 Implementation and Trials

The implementation of the LBPA was largely influenced by two principle considerations concerning the term 'location' and the composition of the messages. Before discussing the application implementation process, these two principles are described.

9.4.1 Design Principles

Location

The term 'location' in our context refers specifically to locations defined in a geographic reference system, e.g., longitude and latitude. Each mobile PinboardClient node always provides information about its current location (e.g., by using a GPS receiver) to the application. This allows the application to compare the current node position with the target area. The target area specified in the messages can be defined either as a rectangle (GeoRectangle), a circle (GeoCircle) or a point (GeoPoint), which are described below.

A *GeoPoint* is a geographic point given by a location descriptor. It is defined unambiguously by latitude, longitude and optionally by elevation within the accuracy of the positioning system.

A *GeoRectangle* is defined by two points p_1, p_2 and a length l_1. p_1 is defined by the location descriptor, which contains numerical values for geographical latitude, longitude and elevation. p_2 and l_1 define the location area descriptor in the following way: A straight line from p_1 to p_2 defines the base line which is identical to one edge of the rectangle. l_1 defines the length taken orthogonal with respect to the base line. For positive values of l_1, the direction of l_1 has to be set clockwise, negative values result in anti clockwise directions.

Last, a *GeoCircle* is defined by a center point p_1 and a radius r giving the distance from the center in meters.

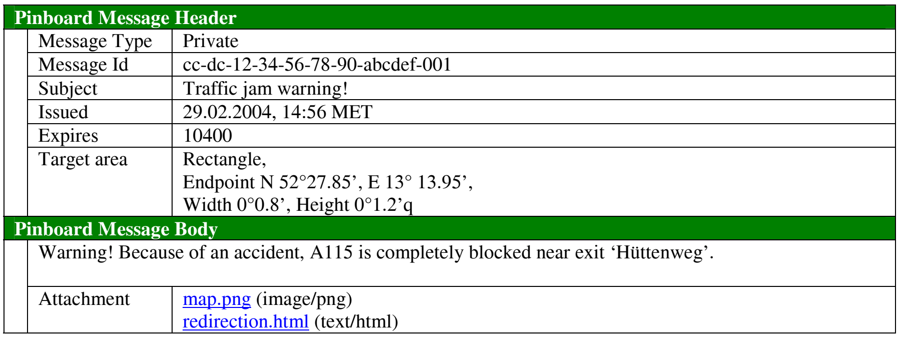
It must be noted, that the definition of location and area as given above described the point of view of the application programmers. On the one hand, it might be possible to represent a desired target area by approximating to a circle or rectangle. On the other hand, the application user must not be constricted by such definitions, the application must allow for areas defined by arbitrary placed points (e.g., derived from a map representation), which could be represented by polygons. Allowing them within the PinBoard location definition will enhance the application capabilities.

Messages

The PinboardMessages are XML-based documents and defined in a comprehensive manner to accommodate for future extensions to the application. The message format defines functionality to support two-way acknowledgments, explicit requests and content broadcasts. A PinBoard XML message consists of a header containing control information and a body containing the message content – an example is given in Figure 9.4.

The message *header* contains all information necessary for processing the message:

- a *message ID*, which is automatically generated and uniquely identifies each message;
- a *subject*, giving an overview of the content of message;
- a *lifespan* and a *creation date*, indicating the start and the end for the validity of the message;
- a *target area* specifying a geographical region as mentioned above; and
- a *message type*, indicating how the message must be handled by the application client.



Pinboard Message Header		
	Message Type	Private
	Message Id	cc-dc-12-34-56-78-90-abcdef-001
	Subject	Traffic jam warning!
	Issued	29.02.2004, 14:56 MET
	Expires	10400
	Target area	Rectangle, Endpoint N 52°27.85’, E 13° 13.95’, Width 0°0.8’, Height 0°1.2’q
Pinboard Message Body		
	Warning! Because of an accident, A115 is completely blocked near exit ‘Hüttenweg’.	
	Attachment	map.png (image/png) redirection.html (text/html)

Fig. 9.4. PinboardMessage example

The message *body* contains a textual part and optionally one or more links to attachments. The links are necessary to optimize bandwidth utilization by transmitting attachments only when the user shows interest by expressly requesting it. The LBPA supports audio, video, image and text attachments.

9.4.2 Implementation

The foundation for the realization of the LBPA was provided by a formal specification, defining use case models, time line models and system class models. However, the purpose of this section is not to fill it with lots of diagrams but to describe solutions to technical challenges and explain decisions taken during the implementation.

The goal of the LBPA implementation process was an early deployment into the FleetNet VANET. Therefore, the functionality was constraint to a subset of the full specification. Special interest was given to the realization of the example scenario described in the previous section. Therefore, the implementation process was divided in three phases: the initial phase, the extended deployment phase and the final integration phase. Figure 9.5, besides describing the application interfaces, also illustrates this phased approach: the upper section encloses the first phase, the realization of the entities in the lower part corresponds to the second phase, whereas the complete picture is the outcome of the last integration phase.

In the first phase, the Internet PinboardClient, the PinboardServer and the PinboardDatabase were implemented as ‘pure’ Internet solutions. Based on the specification, the interface between the PinboardClient and the PinboardServer was realized as a simplification of ‘web services’. Messages were formatted as XML documents and exchanged between PinboardClient and PinboardServer using standard TCP/IP sockets. A sample PinboardMessage containing all relevant information can be seen in figure 9.4. Both PinboardClient and PinboardServer were implemented in Java$^{\text{TM}}$, simplifying the construction of the client interface, the integration of the database and assisting

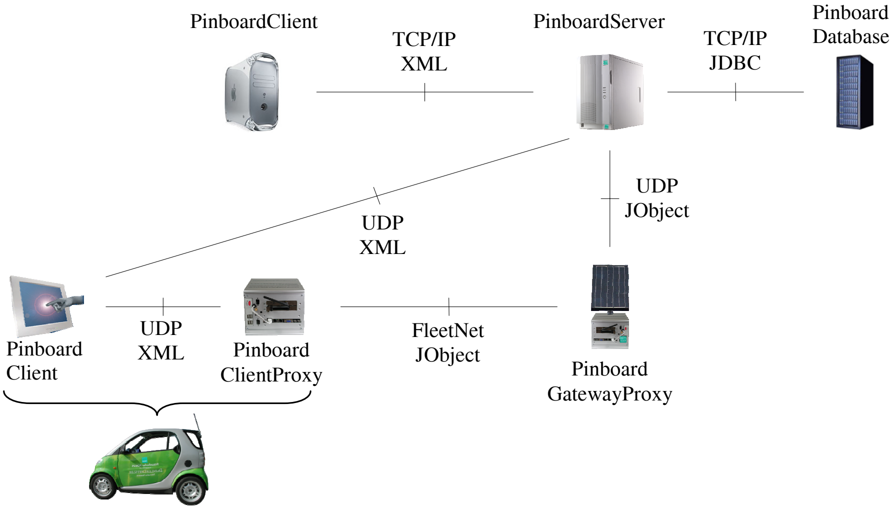


Fig. 9.5. PinBoard interface implementation

in realizing the modular application design. Our first notion to completely embrace Java^{TM} web services by using a Simple Object Access Protocol (SOAP) interface between client and server had to be abandoned because the realization of such a protocol between mobile client and the server within the VANET generated a lot of unnecessary problems. Nevertheless, a SOAP interface to the application can always be provided as an additional module, e.g. a translation thread running on the server platform. The database was implemented using PostgreSQL, offering support for geographical coordinates, geographical indexing and full Java^{TM} Data Base Connectivity (JDBC). The first phase ended by verifying the important interface between the PinboardServer and the PinboardDatabase as well as the communication between PinboardClient and PinboardServer when sending messages.

The second phase extended the implementation to the mobile PinboardClient and the PinboardProxy components: the PinboardProxyClient and the PinboardProxyGateway. The interface between the gateway proxy and the server, still part of the 'wired' infrastructure, was realized by transmitting Java^{TM}objects (JObject) over standard UDP. The gateway proxies implemented stubs to convert all messages received from the PinboardServer to the FleetNet VANET transport protocols facilitating geographic addressing. The client proxies had to reverse the message conversion and format the message back to XML before sending it to the vehicle client. By using this technique, two options became feasible. First, it was possible to replace the underlying network for testing purposes. Both proxy components needed to interface with the VANET, one of the last things to be available for testing and integration. So, the VANET specific stubs were at first replaced by stubs working on standard Internet networks. Second, the client implementation was identical for the Internet client and the mobile client, only the interface had to be adapted to the specific environment. As a result of the second phase, a complete scenario with a simulated VANET connection could be examined and verified.

The third and last phase integrated the LBPA components into the real FleetNet VANET. Having identified most implementation related problems in the previous two phases, the application and the evolving network could both be tested.

9.4.3 Trials

The LBPA passed through a number of trials within the FleetNet framework, ranging from stationary, on-desk tests to full motion scenarios featuring five cars and four FleetNet Gateways (FGWs).

The trial preparation required the installation of wireless equipment and other relevant hardware into the vehicles. Next, the components, especially the FGWs, had to be deployed and the LBPA software had to be installed and tested. The exact geographical location of all stationary nodes was determined and configured. To visualize the results of the tests immediately, the LBPA was extended to interface with a showroom application. This software is an application which can display arbitrary objects on a real map based on their location, as seen in figure 9.6. Next, sample messages were generated, each message having different time ranges and geographical regions. All geographical regions were chosen to be circles to simplify the trial scenarios.

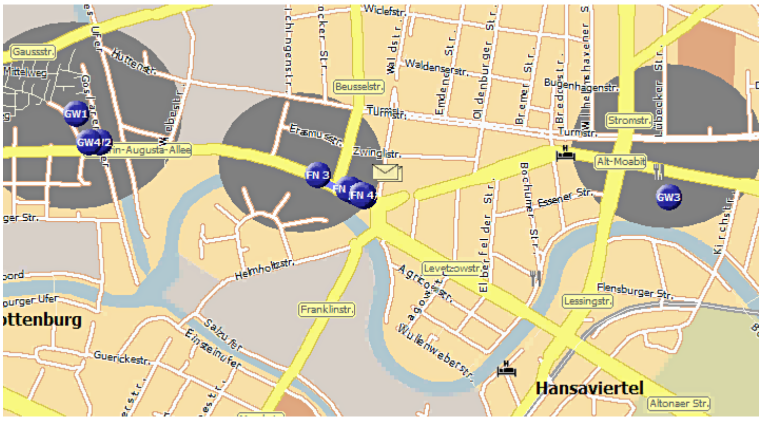


Fig. 9.6. Showroom Application

The trials began by sending the PinboardMessages to the server. At this time, database access tools could verify that the messages were stored within the database. The message distribution from PinboardServer to proxies was checked by monitoring the low level connection to and from the proxies. Next, the mobile nodes – the cars – were sent to a round trip between two locations as seen in figure 9.7. During the trip, two functions could be verified:

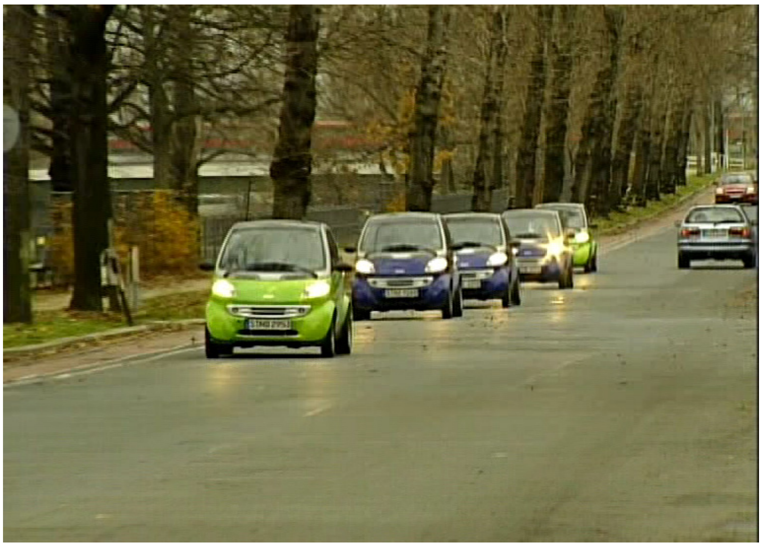


Fig. 9.7. Vehicle Convoy during Trial

- Whenever a vehicle entered the radio coverage of a road-side gateway, new messages were transferred from the FGW to the vehicles and from the vehicles to other vehicles via the VANET.
- Vehicles entering a geographical region of an already received message displayed this message.

Already verified in a lab environment, the real world trials confirmed the correct behavior of the LBPA under extreme conditions – normal, rush-hour traffic. The set-up of such trials is quite complex and takes a lot of resources. Therefore, the frequency of real world trials should be minimized, simulation results should be augmented by lab experiences and, only as a last, expensive step, completed by user trials. The conclusions drawn from the PinBoard trials are listed in the next sections.

9.4.4 Security Considerations

The relevance of security and privacy concerns in Inter-Vehicular Applications can not be overstated. Firstly, as discussed above, there are concerns that IVAs would infringe on and even abuse the personal privacy of the user. In addition, adequate security is a prerequisite for both proper functioning of certain IVAs and for their market acceptance. The current implementation of the LBPA, which was intended as a proof of concept, did not implement security. Nevertheless, we gave considerable thought to the security issues. Our ideas and notions are presented below.

Security can be defined generally as the property of a network or application to provide all expected services, while at the same time providing no hidden services. Specifically, security can be broken down into integrity, authentication, privacy, availability and non-repudiation. A complete analysis of security in IVAs would surpass the scope of this book. Nevertheless, since the underlying VANET is an ad-hoc network, it is important to mention that securing ad-hoc networks is still very problematic and a current topic of research. Therefore, these security concerns are only briefly discussed below along with proposed solutions.

Integrity:

This property states that data must not be corrupted either by transmission errors or by intentional falsification. Should this occur, it must be detected. Messages transmitted within the network must not be altered or falsified. This can be achieved by using checksums and is usually done on the network level.

Authentication:

This security objective refers to the accurate mutual identification of both communication partners or of the source of data. Authenticity is a prerequisite for achieving the other security goals. Pursuing the goal of data integrity for example can only make sense after the communication partners have mutually identified themselves. We advocate a three-fold granular authentication. First, terminal authentication enables the car to authenticate itself to the network before participation in any communication. This can be done using a unique identifier and Public Key Infrastructure (PKI). Factory installed certificates in the communication hardware can be used for the PKI. Secondly, user authentication allows the user to authenticate himself to the application using his user name and password (or Personal Identification Number (PIN)). This authentication information can alternatively be stored on a chip card or an Radio Frequency Identification (RFID) card. User authentication ensures that the correct user profile and personal settings are loaded. Also, messages can not only be traced back to the terminal they originated from, but also to the user who used the terminal. Finally, service authentication is used in conjunction with paid services. It aims to enhance security and is achieved with the help of a Transaction Number (TAN) which is sent along with the request for the service to the server. The server only responds when the TAN is correct and unused.

Confidentiality:

A secure application must ensure that data or resources are made available *only* to authorized entities. This is done using cryptographic tools, PKI and an efficient key exchange mechanism. In a tighter definition, confidentiality requires the network to guarantee that only the entities involved in a transaction should know that the transaction has taken place. This definition is impractical to implement in a VANET since all communications may occur over broadcast radio links.

Availability:

This security objective refers to the constant provision of services and resources by an application and the necessary failure protection mechanisms. Information and resources must be available at the right place, at the right time when needed by authorized entities. The main threats to availability in the LBPA application are Denial of Service (DOS) attacks (jamming the radios of the FGWs, routing table overflow) and network overload using large attachment files.

Non-repudiation:

This security goal refers to the ability to prove indisputably that an entity was the source of the data in question or performed a particular action. A unique identifier is always necessary to achieve non-repudiation. In IVC, a digital version of the Vehicle Identification Number (VIN) is most appropriate. First, the VIN uniquely identifies each car. Secondly, it is usually registered with the legal authorities. Finally, it is difficult to alter. Additionally, altering the VIN is a criminal offense.

Authorization:

While authentication refers to mutually identifying the communication partners, authorization refers to rights management and access controls. A secure application must ensure that only users with the required rights are granted access to network resources or to perform certain actions. This assumes that successful authentication has taken place. The TAN discussed above is a means of rights management.

Trust establishment:

Trust is necessary for the success and acceptance of any technology. If users do not trust the PinBoard messages, they would ignore them and the application would fail. Hence appropriate schemes of trust establishment are necessary. One method is for the server to grant *certificates of credibility* to selected entities. These certificates, provided they are not misused, guarantee the legitimacy of the message. For example, the police and emergency authorities can be issued certificates allowing them to generate and propagate messages on the traffic situation. The certificates of credibility are displayed alongside the messages.

A second method of trust establishment is based on the E-Bay trust model. New users joining the network have a trust rating of *zero*. When they generate messages, they also carry their trust status. Recipients receiving the Pinboard-Message can see the trust status. This gives them an idea of how much they can trust the message. If a message turns out to be correct, recipients are encouraged to give a positive feedback on the message so as to increase the trust rating of the sender. Incorrect messages receive negative feedback, hence reducing the trust rating of the sender. This feedback is sent to the server, which upgrades the senders trust rating by issuing a corresponding *certificate of credibility*. Depending on the size of the network, an appropriate number of positive or negative user reviews are necessary before the trust rating is increased or decreased. In cases of gross misbehavior, the server can decrease the rating to zero. The advantage of this method is that users try to maintain a single identity, because they do not want to loose their reputation. Also, they have a motivation to propagate only valid messages. One of the disadvantages of this proposal is the additional review management overhead it adds to the network. Alternate schemes can also exploit implicit trust relationships from other applications. A far-fetched example is using a users E-bay reputation in the LBPA.

Misbehavior Detection:

False messages do not only lead to a drop in the acceptance of the LBPA, but can also be used in large numbers to mount DOS attacks. Worse still, a false message of a road block can cause unnecessary traffic jams. It is thus necessary for misbehavior to be prevented and quickly identified if it occurs so as to minimize the damage to the network.

The difficulty starts with defining which user activities constitute misbehavior. This definition needs to be broad enough to minimize the threats to the application, while at the same not restricting the functionality of the application. For example, a node propagating messages over the VANET without having a valid VIN can be assumed to be misbehaving, as only vehicles are allowed to transmit messages in that way. Other aspects include using the number of messages sent by a user to determine if he or she is attempting to flood the network.

9.4.5 Open Issues and further Work

As mentioned above, only the core components of the LBPA were implemented. Furthermore, we became aware of certain issues regarding the application and the underlying network during the implementation and trial phases. These issues can be grouped into technical difficulties and areas of utilization.

Technical constraints

The LBPA was developed with the constraints of certain technical realities. These were mainly its dependence on the GPS and the restrictions in handling large files.

Dependency on GPS

In the current implementation, the definition of the *location* is limited by the capabilities of in-car navigation systems, which rely on GPS. However, the accuracy of the location information and the flexibility of the application could be improved by integrating additional sources of location information: the upcoming Galileo system or location data extracted from GSM and/or pico cells in the immediate vicinity of the vehicle. In principle, it is possible to enhance the location management module of the LBPA to support these sources, but the idea was not been further explored.

Managing large files

The second technical concern arises from the broadcast nature of the wireless interface. Although this characteristic is crucial for the proper functioning of the LBPA, the distribution strategies for point-to-point messages and for large message attachments needs to be examined more closely. The air interface in the VANET is quite vulnerable to congestion. The simple transmission of a large data file might render the complete network congested and unusable. It is therefore important to determine whether it is more efficient to broadcast large attachments at once with the message or to transmit just a link to the attachment along with the PinboardMessage and let each user request the attachments as needed. Factors like the car density and number of interested users in the target network will influence the distribution strategy selected.

Area of Utilization

During tests and trials, the LBPA proved to be more than effective in transporting messages within the FleetNet VANET. Foresight in the design of the application makes it a flexible and extensible application for ad-hoc networks. The possible areas of utilization by far exceeds simple messaging as was in the case in the initial testbed. Commercial and other free services can be built on this application platform. Depending on the intended application area, detailed analysis will be necessary.

For example, to use the LBPA as a deployment application in IVC, extensive simulation is necessary to analyze how the application performs in situations of higher node density.

In addition, commercial content providers on top of the LBPA will require an environment allowing service provisioning, charging and billing. Therefore, detailed exploration of transaction safety both in the network (e.g., acknowledgment delivery), and in the application (database and application code) must be undertaken.

9.5 Conclusion

VANETs will be deployed around the world within the next decade both because of compelling applications and since the required technology has become readily available. An early indication for this is that car and equipment manufacturers invest significant attention and money in this area. VANETs will provide a network basis, which cannot be compared in type and scale to any network infrastructure existing today. New applications will be needed to populate the ad-hoc inter-vehicle environment with life, to make the usage of such resources as easy and natural as mobile phones now. Based on the experiences gained in the FleetNet project this chapter provided an entry point to all who are interested in developing future VANET applications.

Acknowledgments

The authors wish to thank all colleagues involved in the FleetNet project and everyone working within the Fraunhofer Institute FOKUS. Special thanks to Bernd Bochow, without his dedication and resourcefulness the realization of this project would not have been possible. Also, we wish to thank Thomas Luckenbach for permitting and encouraging us to work on the PinBoard application.

Acronyms and Abbreviations

BER Bit Error Rate
CBR Constant Bit Rate
CDA Cooperative Driver Assistant
DOS Denial of Service
FGW FleetNet Gateway
GPS Global Positioning System
GUI Graphical User Interface
HCI Human-Computer Interface
IVA Inter-Vehicular Application
IVC Inter-Vehicular Communication
JDBC Java$^{\mathrm{TM}}$ Data Base Connectivity
LAN Local area network
LBPA Location-Based PinBoard Application
MDR Minimum Data Rate
MMI Man-Machine-Interface
MN Mobile Node
MTBF Mean Time Between Failures
MTTR Mean Time To Repair
OEM Original Equipment Manufacturer
PDR Peak Data Rate
PIN Personal Identification Number
PKI Public Key Infrastructure
QoS Quality of Service
RFID Radio Frequency Identification
SDR Sustained Data Rate
SMS Short Message Service
SOAP Simple Object Access Protocol
TAN Transaction Number
TCP Transmission Control Protocol
UDP User Datagram Protocol
VANET Vehicular Ad-hoc Network
VIN Vehicle Identification Number
VUI Vehicle User Interface
WLAN Wireless local area network

References

1. Bisdikian, C., Christensen, J., Davis, J., Ebling, M.R., Hunt, G., Jerome, W., Lei, H., Maes, S., Sow, D.: Enabling location-based applications. In: Proceedings of the 1st international workshop on Mobile commerce, Rome, Italy, ACM Special Interest Group on Mobility of Systems, ACM Digital Library (2001) 38–42

2. Dey, A.K., Abowd, G.D.: Towards a better understanding of context and context-awareness. Technical Report 99-22, Georgia Institute of Technology (1999)
3. Spriestersbach, A., Vogler, H., Lehmann, F., Ziegert, T.: Integrating context information into enterprise applications for the mobile workforce - a case study. In: Proceedings of the 1st international workshop on Mobile commerce, Rome, Italy, ACM Special Interest Group on Mobility of Systems, ACM Digital Library (2001) 55–59
4. Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to context than location. Computers and Graphics **23** (1999) 893–901
5. Marcus, A.: Fast forward: The next revolution: vehicle user interfaces. In: Interactions, Volume 11 Issue 1, Aaron Marcus and Associates (2004) 40–47
6. Sarker, S., Wells, J.D.: Understanding mobile handheld device use and adoption. In: Communications of the ACM, Volume 46 , Issue 12, New York, USA, ACM Special Interest Group on Mobility of Systems, ACM Press (2003) 38–42